FREE INSIDE! Amiga Product Guide: Hardware Edition

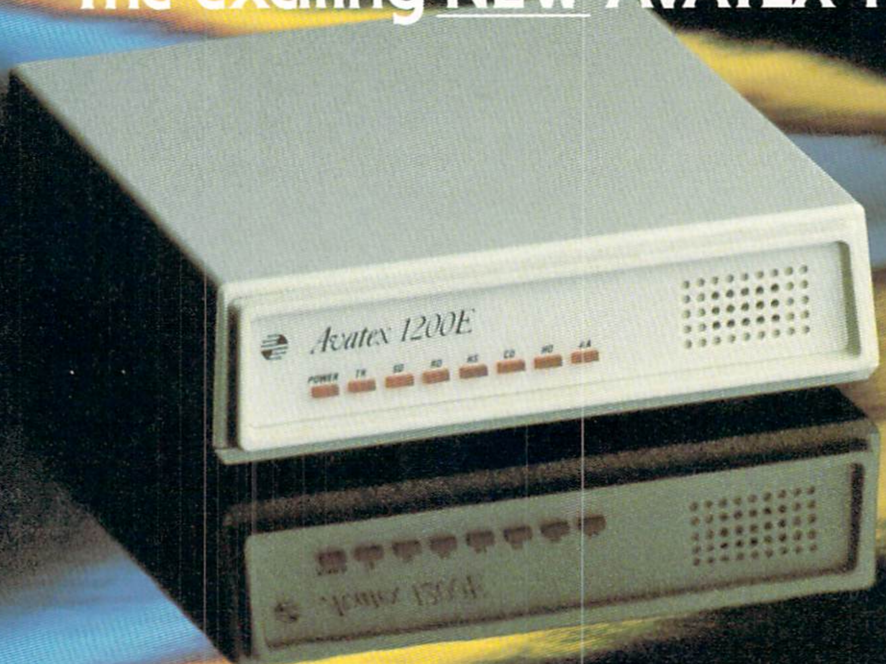
Amazing COMPUTING™

Your Original AMIGA™ Monthly Resource

Volume 3 Number 5
US $3.95 Canada $4.95

VIDEOSCAPE 2.0!

Getting Started with CLI
Compilers in the Public Domain



GOOD THINGS COME IN SMALLER PACKAGES.
The exciting NEW AVATEX 1200 E.

$85
TOTAL DELIVERED PRICE.

AVATEX
Authorized Sales &
Service

ACTUAL SIZE ONLY 5" X 6"

FREE! Communication Software & Compuserve Access Time with each MODEM.

We give you a great price . . .

- + TOLL FREE ASSISTANCE
- + IMMEDIATE DELIVERY
- + NO CREDIT CARD SURCHARGE
- + NO SERVICE OR HANDLING FEES

+

FREE SHIPPING!

TOTAL HAYES COMPATIBILITY
8 LED STATUS INDICATORS
AUTO DIAL AND ANSWER
TONE OR PULSE DIALING

NEW 1200 E FEATURES

CCITT COMPATIBILITY
CALL PROGRESS DETECTION
INTERNAL SPEAKER
2 YEAR WARRANTY

Thousands of customers have purchased the AVATEX 1200 from Megatronics. New technologies now make it possible to offer an improved version of this popular Modem.

The 1200 E is amazing.
It weighs less and is smaller.
It has even more features.
It's still the same low price.

MEGATRONICS

MEGATRONICS, INC. P.O. BOX 3660, LOGAN, UT 84321

To Order
CALL FOR FREE CATALOGUE
CREDIT CARDS VERIFIED FOR YOUR PROTECTION

800-232-6342
INSIDE UTAH (801) 752-2642 FAX (801) 752-8752

We'll beat any advertised price.
MEGATRONICS, INC., P.O. BOX 3660, LOGAN, UTAH 84321

Go ahead... Make Your Day!

Now you can shoot the bad guys with this real-time action shoot-'em-up adventure. Just connect the Actionware PHASER™ to the game port (or use your mouse) and you're ready to combat evil in an exciting action packed world!

It's your choice . . . ***CAPONE***™ gangsters in Chicago, ***P.O.W.***™ enemies in Asia, ***CREATURE***™ aliens aboard your spaceship.

Each Action Adventure only $39.95
Actionware PHASER (optional) $49.95

See your Dealer or
call **1-800 848-2333** to order
In Illinois (312) 879-8998

VISA/Master Card accepted

Actionware Corporation
38 W 255 Deerpath Road Batavia, Illinois 60510
AVAILABLE ONLY FOR THE AMIGA WHICH IS A TRADEMARK OF COMMODORE-AMIGA

Amazing COMPUTING

Volume 3, Number 5

CONTENTS

Amazing Features

Interactive Startup Sequence by Udo Pernisz 12
Forge an interactive relationship with your Startup-Sequence.

The Command Line by Rich Falconburg 27
There's no need to fear the Amiga Command Line Interface.

AmigaTrix III by Warren Block 36
Tips and tidbits to make your Amiga life a whole lot easier.

Amiga Product Guide: Hardware Edition 55
A descriptive listing: Over 250 Amiga hardware products!

Proletariat Programming by Patrick Quaid 75
What's your best bet for a public domain compiler? Give a listen.

The Companion by Paul Gosselin 89
Tinkering with the Amiga's Event Handling capability.

Amazing Columns

Take Five! by Steve Hull 16
The Game of the Month is . . .

Hot on the Shelves! by Michael T. Cabral 33
A new column uncovers the hottest new products.

Bug Bytes by John Steiner 53
BZZZZZZZZZZZZZZZ. Splat.

The Big Picture by Warren Ring 83
The three-part Unified Field Theory winds up with source code and examples.

Modula-2 by Steve Faiwiszewski 102
Termination modules for Benchmark and TDI compilers.

68000 Assembly Language by Chris Martin 111
Peeling away the complication of display routines.

Roomers by The Bandito 115
Commodore at NCGA, the A2024 monitor, & more....

In the Public Domain by C.W. Flatte 117
The Fred Fish collection hits 138! Here's a look inside the first two new entries.

Amazing Reviews

MindLight 7 by David N. Blank 8
Psychedelic fad of the 70's updated for the Amiga.

VideoScape 3-D 2.0 by David Hopkins 24
Push desktop video to the limit

Extend by Bryan D. Catley 43
An AmigaBASIC extension that brings many of those slippery Intuition routines within your reach.

AssemPro by Stephen Kemp 47
Opening a door to assembly language programming.

APL.68000 by Roger Nelson 49
"A Programming Language" comes to the Amiga, complete with all its ease of use and efficiency.

Book Reviews by Richard Grace 109
"C" between the covers of three programming texts.

CBTREE by Michael Listman 113
A tidy collection of functions to aid the C programmer.

Amazing Departments

Amazing Mail 4
Index of Advertisers / Reader Service Card CIII
Public Domain Software Catalog 120

Other Products from The Other Guys

REASON - a professional proofreading system used by universities and writers around the world to analyze and improve writing. **(Has helped raise students grades when used faithfully.)** **$395.00**

OMEGA FILE - a *REAL* data base & mail merge	**$79.99**
PROMISE - the *BEST* high speed spell checker. (Even better than Zing!® Spell)	**$49.99**
KEEP-Trak GL - general ledger for home or business	**$49.99**
AMT - amortization program	**$39.99**
MATCH-IT -teaches shapes & colors (preschool)	**$39.99**
MATH-A-MAGICIAN - add, subtract, multiply & divide	**$39.99**

Call or write for more information.

SYNTHIA *High Performance Digital Synthesizer*

A state of the art music tool which will:
Create digital **IFF Instruments** for use with nearly all music programs!
Modifying existing **IFF Instruments.** Use *SYNTHIA* on digitized samples to add reverb, wow, and other enhancements.

SOMETHING FOR EVERYONE:

Additive Synthesis - a traditional method which can create almost any type of instrument.
Plucked String Synthesis - simulates plucked strings . . . right down to the 'pluck'.
Interpolative Synthesis - a method which introduces the natural imperfections found in instruments. (Instruments such as brass, woodwinds, pianos, etc.)
Percussion - build your own drum set . . . create any drum you desire.
Subtractive Synthesis - a simple method of creating instruments.
Special Effects - includes filtering, amplification, phasing, waveshaping, amplitude modulation, real reverb, and . . .
IFF Music Player - powerful and compact. Now you can enjoy those songs that needed a memory expansion before! Up to 32 tracks and 32 IFF Instruments! Supports chords, ties, etc.

NOW SHIPPING

IS IT LIVE . . . OR IS IT SYNTHIA?

Synthia uses the latest technology to generate realistic sounding instruments and even the new families of instruments sound real. A real synthesizer on a real computer!
Why buy digitized instruments when you can *SYNTHIAsize* them?

Requires AMIGA 512K

$99.99

Copyright©1987, THE OTHER GUYS Software • AMIGA is a registered trademark of Commodore Amiga

THE OTHER GUYS

55 North Main Street
Suite 301-D
PO Box H
Logan Utah 84321

(801) 753-7620
(800) 942-9402

AMAZING MAIL

Amiga Users: Beginners and Advanced!

Dear Amazing Computing,
I have been an avid reader since Volume 1.1 and still find AC to be the number 1 Amiga magazine.

Your articles on Assembly (machine) language programming are most appreciated. I have been programming in BASIC and the appearance of Machine Language articles helped me make the decision to go with ML rather than "C."

The purpose of this letter is not just to praise your efforts but to contribute a little knowledge that I have acquired. In Volume 2.4, Mr. Greg Douglas presented a short article on using function keys with MicroEmacs, as supplied on Extras 1.2. I finally got around to setting up an Emacs disk and attempted to create a BASIC program to accomplish what Mr. Douglas had done with his C program, build a custom file that Emacs would load.

Instead, in re-reading the Emacs documentation, I discovered the method for creating the config/emacs_config file from within MicroEmacs.

The Quote Char option under the Edit menu is the key. Anytime you need to enter a control sequence first select the Quote Char option then type the control sequence. The Quote Char option is only active for one control sequence, that is if you wanted to enter ^A^K for a specific function key, you would select Quote Char then type ^A, reselect Quote Char and type the ^K. (Control is represented by ^).

Once you have defined all 24 keys F1-F10, Help, 0-9,.,-, and ENTER, simply use the Save As function and save the Emacs file as config/emacs_config in your root directory and the key map will be loaded when Emacs starts.

You could also use this same method and save other key maps and load them after Emacs is started with the "Load Keys" option.

One other error has been discovered in the documentation. It states that ^J will perform an indent function while in reality it simply opens a blank line in your text.

A warning, I was visiting files and creating files during this learning process and discovered that if you run out of memory for Emacs buffers you will get an "error in memory allocation" message and the only way out is to re-boot! Other tasks, Workbench, Etc. still seem to be performing normally, but Emacs is definitely in trouble.

I hope this bit of data will be of some assistance to other new MicroEmacs users.

Sincerely,
Jim Thompson

Dear Amazing Computing:
As a new Amiga owner and a recent subscriber to your magazine, I find that I'm enjoying both very much. I particularly enjoy your articles on BASIC and your reviews of public domain programs. From my viewpoint as a beginner at all of this, I have a couple of suggestions which I think may make your magazine even more useful to a broader readership.

1: For greater ease and accuracy in entering type-in BASIC programs, a convention is needed to indicate the number of spaces to be entered at any given point in the program. Perhaps a box with a number in it to indicate the number of spaces to be entered would be appropriate. This would save a great deal of time and effort on the reader's part trying to count just how many spaces there are in a "blank" part of the code, particularly when the code is formatted so that the spaces go from one line to the next.

2: For a better understanding of the programs on the Amicus and Fish disks, a small article each month describing a selected program in some detail would be most useful. The article could include how to install the program, bugs to be alert for, and how to use it to its maximum potential. I have found many interesting programs on the disks but have spent hours trying to get some of them running when there has been sketchy documentation or none at all.

Yours truly,
John M. Salimbene
Ohio

We appreciate your warm words and both of your suggestions are well received.

Amazing COMPUTING

ADMINISTRATION

Publisher:	Joyce Hicks
Assistant Publisher:	Robert J. Hicks
Circulation Manager:	Doris Gamble
Asst. Circulation Mgr:	Traci Desmarais
Corporate Trainer:	Virginia Terry Hicks
Traffic Manager:	Robert Gamble

EDITORIAL

Managing Editor:	Don Hicks
Co-Editor:	Ernest P.Viveiros Jr.
Co-Editor:	Michael T. Cabral
Hardware Editor:	Ernest P. Viveiros Sr.
Music & Sound Editor:	Richard Rae
Amicus & PDS Editor:	John Foust
Copy Editors:	Julie Landry Michael Creeden Albert G. Andrade

PRODUCTION

Art Director:	Keith Conforti
Illustrator:	Brian Fox
Production Manager:	Mark Thibault
Associate Prod. Mgr:	Rico A. Conforti

ADVERTISING SALES
Advertising Manager: John D. Fastino
1-800-345-3360 or
1-617-678-4200

SPECIAL THANKS TO:
Buddy Terrell & Byrd Press
Betsy Piper at Tech Plus
Bob at Riverside Art, Ltd.

Amazing Computing™ (ISSN 0886-9480) is published monthly by PiM Publications, Inc., P.O. Box 869, Fall River, MA 02722-0969. Subscriptions in the U.S., 12 issues for **$24.00**; in Canada & Mexico surface, **$36.00**; foreign surface for **$44.00**. Application to Mail at Second-Class Postage Rates pending at Fall River, Ma. and additional mailing offices. **POSTMASTER**: Send address changes to PiM Publications Inc., P.O. Box 869, Fall River, MA 02722-0969. Printed in the U.S.A.

Copyright© April1988 by Pim Publications, Inc. All rights reserved. First Class or Air Mail rates available upon request. PiM Publications, Inc. maintains the right to refuse any advertising. Pim Publications Inc. is not obligated to return unsolicited materials. All requested returns must be received with a Self Addressed Stamped Mailer. Send article submissions in both manuscript and disk format to the Co-Editor. Requests for Author's Guides should be directed to the address listed above.

Amazing Dealers

The following are Amazing Dealers, dedicated to supporting the Commodore-Amiga™. They carry Amazing Computing™, your resource for information on the Amiga™. If you are not an Amazing Dealer, but would like to become one, call PiM Publications Inc:

1-617-678-4200

Alaska	
That Computer Place	Anchorage
Alabama	
Abax Data Systems	Huntsville
Alabama Book Store	Tuscaloosa
Command Computers	Birmingham
Madison Book & Computer	Madison
Mel's Photo & Computer Shop	Montgomery
Universal Computer System	Mobile
Arkansas	
Computermart	Mountain Home
The Micro Shop	Little Rock
Arizona	
Commodore Connection	Sherwood
Computer Libray, Inc.	Tempe
Computer Works	Glendale
Copperstate Business Systems	Phoenix
Hacko's	Tempe
North American Digital	Tucson
California	
Aday International	Foster City
Augnet	Fremont
Balboa Office Supply	San Diego
Bits n Pieces Comp Prod.	Anaheim
Brown Knows Computing	Redlands
Byte Factory Computers	Salings
Century Computer Systems	La Habra
Circus of Books	Los Angeles
Circus of Books	Los Angeles
Commodore Land	Alhambra
Commotech	National City
Compunik	Los Angeles
Computer Attic	Palo Alto
Computer Cellar	Anahiem
Computer Nook, Inc.	San Bernadino
Computer Outlet	San Diego
Computer Outlet	San Marcus
Computer Spectrum	
Computer Time	Citrus Hts
ComputerLand Fremont	Fremont
Computer Literacy	San Jose
Coronet Newsstand	Oceanside
Dataphile	Watsonville
Delta Spectrum	Laguna Beach
Delauer's News Agency	Oakland
Discount Computer Station Inc.	Long Beach
Empire Computers	Santa Rosa
ECX Computer Co.	Walnut Creek
Fisher Computers + Software	Rancho Cucamonga
Harding Way News	Stockton
Heinrichson, Schneider & Young, Inc.	San Francisco
Home Computing Center	Concord
Home Computing Center	Mill Valley
Home Computing Center	San Leandro
Home Computing Center	San Bruno
HT Electronics	Sunnyvale
International Telecine	Santa Ana
K J Computers	Granada Hills
Lamar Micro	Redondo Beach
LTC Computers	Santa Ana
Micro Flight	San Jacinto
MTS Inc. Tower Books	Sacremento
Nonagon Software	El Cerrito
Op Amp Tech Books	Los Angeles
Paras Newsstand	San Diego
Put's Electronics	Rio Linda
R&R Software	Chula Vista
Real Computers & Electronics	Petaluma
Ridgecrest Computer Ctr. Inc.	Ridgecrest
Sawyer's News Inc.	Santa Rosa
S.O.S. Computers	Los Angeles
Software First	Santa Rosa
Software Cellar	Anahiem
Software Central	Pasadena
Software Plus	Citrus Hieghts
Software Sollutions Plus	El Cerrito
The Booksmith	San Francisco
Tower Books	Chico
Tower Books	Citrus Heights
Tower Books	Concord
Tower Books	Mountain View
Tower Books	Sacremento
Tower Books	Sacremento
Tower Books	Sacremento
Tower Books	San Mateo
The Booksmith	San Francisco
The Computer Room	Scotts Valley
WCI Computer Central	Eureka
Warner Engineering	San Diego
Winner's Circle	Berkley
World Book & News	Hoolywood
Colorado	
Aurora Newsstand	Aurora
Book Tech Distributing	Englewood
Citadel Computers	Colorado Springs
Computer Discount	Denver
Computer Repeats	Boulder
Computer Room	Aurora
Ebert Personal Computers	Aurora
Micro Computer World	Fort Collins
Micro World	Lakewood
Connecticut	
Aetna Life Club Store	Hartford
Amigo Business Computer	Norwalk
Computer Outlet	PlaVernon
Friendly Computers	Plantsville
Mnemonics Inc.	Norwalk
Personal Computer Center	Norwich
Software City	Orange
Software Kingdom	East Windsor
Tricom Computers	North Windham
Delaware	
Castle Video	Newark
Nor-Mar Inc.	Wilmington
Florida	
A A Computers	Jacksonville
Commodore Country Inc.	Pinellas Park
Computer Bar	Pennsacola
Computer Base	Brandon
Computer Image	Miami
Computer Room	Ft. Lauderdale
Computer Specialty	Melbourne
Computer Terminal Inc.	Sebring
Computers 4 Rent	W. Palm Beach
Computers Plus	S. Daytona
Cotronics Inc.	Stewart
Education Computers Etc.	Talahasse
Family Computers	Sarasota
Florida Book Store	Gainesville
Gulf Coast Computer Exchange	Panama City
Heartland Computers	Mulberry
Joe's News, Inc.	Miami
Micro Computer Center	Largo
New Age Electronics	St. Petersburg
Program Store	Clearwater
Random Access Computers Inc.	Ft. Walton Beach
Software City	Lauderdale Lakes
Software City	Orange Park
Software Software	Plantation
Sunny's At Sunrise, Inc.	Sunrise
The Computer Store	Lakeland
The Open Door	Cocoa
The PC Collection	Coral Gables
Georgia	
BFJ Enterprises	Roswell
Future System	Augusta
Software South Inc.	Savannah
The 64 Store	Atlanta
Hawaii	
Astro Computers & Electronics	Aleia
Computer House Inc.	Honolulu
Data-OneComputer Center	Honolulu
J & C Repair	Alea
Idaho	
A.B.I. Computer & Video	Nampa
Beacon Micro Center	Ames
Boise Computer & Video Inc.	Boise
Illinois	
A Bitcketer	Rockford
Computer Cellar	St. Charles
DigitalWorld	Addison
DOS Computer Center	Niles
I.C.V. Data Center Inc.	Mundelein
Keepin Pace Computers	Champaign
Kroch's & Brentano's	Chicago
Micro Age	Oaklawn
Microtek	Kankakee
Software City	Belleville
Software Or Systems	Fairview Hts.
Software Plus	Wheeling
Software Plus Chicago	Chicago
Software Plus West	Hanover Park
The Memory Expansion	Bloomington
Warren Radio	Peoria
Unique Computer	Sterling
Indiana	
Burkat Computer Ctr.	South Bend
Bytrex of Angola	Angola
Computer Corner	Fort Wayne
Computer Corner	Marion
Computer People Inc.	Michigan City
CPU Inc.	Indianapolis
Indiana News	Indianapolis
Micro Computer Centers Inc.	Indianapolis
Reese Equipment Co.	Plymouth
Spectra Computer Sys. Inc.	Evansville
The Book Corner	Blumington
The Computer Room of Merrillvile	Merrillvile
Vons Computers	West Lafayette
Iowa	
Gronet Computers	Des Moine
Micro Computer Applications LTD	Marshalltown
Soft Search	Ames
Kansas	
J & L Electronics	Liberal
MidKansas Computers	Newton
Midwest Computers	Manhattan
Software City	Overland Park
Kentucky	
Computer Store of Kentucky	Louisville
Hawley Cooke	Louisville
Louisiana	
Computer Time	Shreveport
Modern Business Machines	Metairie
Software Center International	Metairie
Software Source	Lafayette
The Bok Shelf	Slidell
The Computer Clinic	Lafayette
Maryland	
Buried Treasure	Rockville
Cal Comp Inc.	Silver Springs
Compuvision Vision Center	Columbia
Greetings and Readings	Towson
Knowledgeware Systems	Kensington
Layhill Newsstand	Silver Spring
Micro Computer Center	Baltimore
Professional Micro Services	Baltimore
Software Advantage	Rockville
Software N Things	College Park
Waldorf Computer	Waldorf
Massachusetts	
Club Computer	Cambridge
DOX	Wiliamstown
General Computer Store	Framingham
HCS Computer Center	Marshfield
LCA Video & Computer Ctr.	Norwood
Memory Location	Wellsley
OmniTek Computers	Tewksbury
Pioneer Valley Data Equipment	Amherst
Softpro	Burlington
The Bit Bucket	West Newton
The Software Haus	Quincy
Tycom Inc.	Pittsfield
Maine	
Computer Barn	Damariscotta
Hands On Computing	Westbrook
Radio Shack Associate Stores	Jay
Seaside System / Main TV	Bucksport
Skylight Software	Belfast
Michigan	
Basic Computer Center	Grand Rapids
CompuSoft	Kalamazoo
Direct Access	Lansing
Global Computer Center	Dearborn
Central Park Computers	Okemos
Computer Emporium	Lansing
I/O Software	Flint
Little Professor Book	Dearborn
Michigan Software	Novi
Midland Computer Shop	Midland
Readmore Newscenter	Lansing
Roseville Computer Store	Warren
Slipped Disk	Madison Heights
Specialists Computer	Livomia
State Street Computers	Ann Arbor
T.J. Enterprises	Clinton
The Edge Connector	Cilo
United Computer	Canton
Video Axis	Sterling Hts.
Ye Olde Computer Shoppe	Ypsilanti
Minnesota	
Computers, etc.	Eagan
Computer Exercise World	Minneapolis
Computer Stuff	Minneapolis
JMH Software	Maple Grove
Lookahead Software Computer	Northfield
MCD of Hibbing Inc.	Hibbing
Midwest Computer Services	Clearwater
Shinder's Read More	Minneapolis
Specialists IN	Hopkins
Valiant Inc.	Stillwater
Mississippi	
Computer Connection	Southaven
National Computer Center	Ocean Springs
Missouri	
Associated Computer Services	Springfield
Brands Mart Computers	Kansas City
Data Byte Computer Co.	St. Louis
Instant Replay Ltd.	Chesterfield
Systems Plus	Independence
Systems Plus of St. Louis	St. Louis
Montana	
Appelgren Computer Systems	Great Falls
Nebraska	
BULLS-EYE	Lincoln
My Com Inc.	Plattsmouth
Double E Electronics	Omaha
Nevada	
Century 23 Inc.	Las Vegas
Computer World	Las Vegas
Tower Records	Las Vegas
New Hampshire	
Sitek Inc.	Nashua
New Jersey	
Family Computer Centres	South Orange
Morris County Stationers	Chester
Program Store	Eatontown
Take One More Byte	Hoboken
New Mexico	
New Horizon Computer System	Alamogordo
Page One Newsstand	Albuquerque
Road Runner	Espanola
Technological Concepts	Las Cruces
New York	
Amuse	New York
Amicus- Hudson Valley	Wappingers Falls
Amigo Business Computer	East North Port
Barnes & Noble	New York
Byte Shop	Merrick
Compurite Center	White Plains
Computer Outlet	Jamestown
Computers Etc.	Syracuse
Cromland Inc.	Brooklyn
Leigh's Computers	New York
Mnemonics	North Port
N.U.B.S. Computer Center	Bronx
Softwaire Centre International	Forest Hills
Software & Such	Scotia
Software City	Forest Hills
Software Link	White Plains
Software Supermarket	Tonawanda
Star Tech Systems	Massena
Syntaire Trading	New York
Sysut Computers	Brooklyn
Tadiran	New York
Tedrow Business Products	Rochester
The Computer Cellar	Albany
The Working Computer	Stoney Brook
Unicom Electronics	Johnson City
Video Computer Bin	Brooklyn
Video Computer Center	Rome
Video Connection	Schenectady
Village Computer Inc.	New York
World Computers Inc.	Hicksville
North Carolina	
C2 of Hickory	Hickory
Digitz	Raleigh
K&S Newsstand	Kernersville
K&S Newsstand	Walkertown
K&S Newsstand	Winston-Salem
K&S Newsstand	Winston-Salem
K&S Newsstand	Winston-Salem
News Center	Cary Village
Software City	Charlotte
TDS Computers	Durham
Triad Computers	Winston-Salem
North Dakota	
Computer Associates	Fargo
G. F. Computers	Grand Forks
The Computer Store	Minot
Ohio	
Computer Network	Newark
Computer Site	Cleveland
Earthrise Micro Systems	Columbus
Fairborn Home Computer	Dayton
Lakewood International News	Lakewood
Lakes Consumer Electronics	Akron
Micro Center	Columbus
Microtech Computer Center Inc.	Toledo
Microwave Magic	Fairfield
North Coast Programming	Willoughby
Personal Computer Co.	Youngstown
Paper & Ink	Tiffin
Software and More	Cincinnati
Software City	Centerville
Software Connection	Mayfield Hts.
Oklahoma	
Micro Link Computer Store	Tulsa
P C Tech	Tulsa
Second Hand Software	Oklahoma City
Video-Comp Inc.	Lawton
Oregon	
Andromeda	LaGrande
Capitol News Center	Salem
Clackamas Computers	Beaverton
Comm-Shack	Salem
Computron Business Systems	Portland
Powell's Books	Portland
Powell's Tech	Portland
Rich Cigar Store	Portland
Software Express	Eugene
Software Pipeline	Portland
Pennsylvania	
Alpha Omega EBE	Warren
Basic Computer Systems	Hermitage
Chester County Book Co.	West Chester
Circle Computer	Ephrata
Computers & Games	Reading
Computer Wurx	Stroudsburg
Downington Computer Center	Downington
East Coast Software	Hershey
Floppy Disk	Mechanicsburg
Gene's Books	King of Prussia
North Central Digital Systems	Danville
Pittsburgh Computer Store	Pittsburgh
Tasco Computer Center	Lancaster
Triangle Computers	Indiana
Walden Hill Images	Berlin
Rhode Island	
Micro Limits	Smithfield
Micro Limits	Warwick
Software Connection	Warwick
South Carolina	
Avcom International Inc.	West Columbia
Micro Computer Depot	Sumter
Software City	Spartanburg
Softerware Haus Inc.	Charleston Hts.
Watson's Jewlery	Chesterfield
South Dakota	
Computer Business Center	Rapid City
Tennessee	
Computer Hardware Service	Nashville
ComputerLab of Memphis	Memphis
The 64 Store	Bartlett
Texas	
Amazing Computer Systems	Carrollton
Bookstop	Houston
Byte Shop	Austin
C.E.M. Corporation	Brazoria
Colonel Video	Houston
Computer Age	Houston
Computer Magic	Austin
FNS Computer Services	Texarkana
Inter Computing Inc.	Grande Prarie
Metropolitan Computer	Richardson
Micro Search	Houston
Software Library	Witchita Falls
Software Terminal	Fort Worth
The Computer Experience	San Antonio
The Computer Stop	Abilene
Utah	
Armadillo Brothers	Salt Lake City
Magazines and Gifts	Salt Lake City
Micro Mania	Salt Lake City
People's Computers	Provo
Vermont	
Ray Supply Inc.	South Burlington
Virginia	
Alonso	Alexandria
Diskovery	Falls Church
Family Computer Center	Fairfax
I/O Computer	Norfolk
Software Central	Norfolk
Software City	Richmond
Virginia Micro Systems	Woodbridge
Washington	
Amazing Electronics	Spokane
A.P.P.L.E. CO-OP	Renton
Bellingham Computer Center	Bellingham
Bits, Bytes and Nibbles	Spokane
Bull Dog News	Seattle
Cameo Amiga Sig	Madison
Computer Exchange Services	Bellingham
Computer Quest	Edmonds
Computers + Inc.	Tacoma
Computers and Movie Magic	Yakima
Family Computers	Lynwood
KAOS	Northeast Poulsbo
Microage Computer Store	Bellevue
Nybbles & Bytes	Tacoma
Omni International	Seattle
Programs Plus	Seattle
Real Time Systems	Seattle
The 68000 Club	Spokane
Take A Byte	Kennewick
Tower Books	Bellview
Tower Books	Seattle
Tower Books	Tacoma
Wisconsin	
Computer Software Center	Milwaukee
Fox Valley Personal Computer	Oshkosh
Mom's Computers	Stevens Point
Richard Resle	Madison
River City Software	La Crosse
TMW Software Inc.	Wausau
Wyoming	
Byte Emporium	Laramie
Canada	
Compute or Play	Alberta
Computerworks	Alberta
Computer Shop	Alberta
Data Pro	Alberta
Computer Shop of Calgary	Alberta
The Software House	Alberta
Hindsin Computing	Alberta
Red Deer Micro Systems	Alberta
Memory Land	Alberta
Computer King	British Columbia
Conti	British Columbia
Nutec	British Columbia
Sprita	British Columbia
Strider	British Columbia
Super Software	British Columbia
Canadian Computer Era	Manitoba
Micromart	Manitoba
ComputerWorld	Manitoba
J.W. Allen	Manitoba
Kobetak	Manitoba
Minicomp	Manitoba
Arkon	Ontario
Chatam Computer	Ontario
Computerland	Ontario
Comspec	Ontario
Electronics 2001	Ontario
Hastings Data	Ontario
The Computer & You	Ontario
Thornhill	Ontario
Software Co.	Quebec
Deltronic Solare	Quebec
Mierodata	Quebec
Info Mereier	Quebec
Laval Micro	Quebec
Vision Tronique	Quebec
Distribution Vsipro	Quebec
Distributed in Canada by	
Ingram Software Distributors	Concord, Ontario
Compulit Distributors	British Columbia
Phase 4 Dist. Inc.	Calgary, Alberta
Argentina	
Logic Computer	Buenos Aires
Australia	
Computermart	Leederville,WA
Diskworks	Greenwich, NSW
High Technology Computer Supplies	Brighton, Melborn
John Verhoven	Cloverdale, WA
Maxwell Office Equipment	Abbotsford, Melbum
Megadisk	Lane Cove
Megatronics	Lanecove,NSW
Puerto Rico	
Micro Computer Store of Perto Rico	
Saudia Arabia	
Abdulla Fouad & Sons	Daman
Scotland	
The Amiga Center of Scotland	Edinborough
Sweden	
Nicholas Pearson	Lund
Switzerland	
Software Land AG	Zuerich

Amazing Computing™ is also available in most B. Dalton Booksellers, B. Dalton Software Stores, Crown Books, and Software Etc. locations.

The "blank space problem" is tricky. We use a non-proportional type so all spaces and characters remain in a 1:1 ratio. Adding special blocks to an already crowded code will be very difficult. As a second option, we will return to our previous practice of placing our published code on public domain software disks. We intentionally keep the cost of these disks low, and we hope their inclusion in the public domain will allow a greater number of users an opportunity to not only use the code as published, but to generate modifications and send them back to us for future disks.

On the second note, our new Public Domain Software editor, C. W. Flatte, is attempting to do just as you have asked. He will not only be looking at the newer additions, but he will continue to review older disks and their contents. Special instructions required for the more complicated problems will be an added asset to this column.

Dear Amazing Computing,
This is for those neophytes, like myself, who have met with little success using the IEEE Double Precision library routines under Lattice 3.03.

The C-DEVEL: include/libraries directory contains mathffp.h, which declares the mathffp. library functions. However, the double precision routines are not declared. The fix is to create a header file for the mathieeedoubbas.library called "mathieeedoubbas.h". The file should include the following listing:

```
#ifndef LIBRARIES_MATHIEEEDOUBBAS_H
  #define LIBRARIES_MATHIEEEDOUBBAS_H
  /****************************
  *         mathieeedoubbas.h
  *
  ****************************/

  int            IEEDPFix();
  double IEEEDPFit();
  int            IEEEDPCmp();
  int            IEEEDPTst();
  double IEEEDPAbs();
  double IEEEDPNeg();
  double IEEEDPAdd():
  double IEEEDPSub();
  double IEEEDPMul();
  double IEEEDPDiv();

  #endif LIBRARIES_MATHIEEEDOUBBAS_H
```

This declares the functions listed on pages 472-473 of the ROM Kernal Manual. Place this file in the C_DEVEL:include/libraries directory.

To use the functions, place "#include <libraries/mathieeedoubbas.h>" in the include section of your program and open the library as outlined in the ROM Kernal Manual. The mathieeedoubbas.library file, which can be found in Workbench1.2:libs, should be included in your system disk's libs directory.

The routines are faster and more precise than the standard "C" versions of double precision arithmetic. If you do a lot of double precision number crunching, try out these routines.

Tom Grimes
Texas

Thanks for the tip.

Amiga User Groups

Dear Amazing Computing:
AMiga Users of Calgary (AMUC) would like to invite everyone to visit the Fly By Wire (FBW) BBS at (403) 247-1728. The FBW DDS is available 24 hours a day, seven days a week, and supports 300/1200/2400 baud, 8 bits, no parity, one stop bit.

You can contact AMUC at:
AMUC
P.O. Box 154
Station G
Calgary, Alberta,Canada T3A 2G2

Memberships are $25.00 in Calgary, $50.00 outside Calgary and $60.00 outside Canada (all prices in Canadian funds). Each member receives our monthly newsletter disk. We are interested in swapping newsletter disks with other Amiga user groups.

Yours truly,
Randall G. Rude
AMUC correspondant

Dear Amazing Computing:
I represent L.I.C.A. (Long Island Commodore Amigians) users group which was founded in November of 1986. We are a group of 150 plus persons from Long Island and its surrounding areas. Recently we moved into larger quarters, hoping to expand our membership.

Currently, we are meeting at the State University of New York at Old Westbury. We have available an auditorium which seats 600 people, a large screen projector and a public address system. We meet bimonthly on the first Wednesday and third Friday of every month. At our meetings we have demonstrations, guest speakers representing both software distributors and hardware manufacturers, special interest groups and tutorials. Our public domain library contains over 40 disks, which grow in number each meeting. We produce a monthly newsletter which is mailed to our members, and are currently looking into the feasibility of starting a B.B.S.

L.I.C.A.
(Long Island Commodore Amigians)
Meetings are the 1st Wednesday & 3rd Friday of every month at 8:00 p.m. in Room D-190 (auditorium) of S.U.N.Y. Old Westbury (Route 107)
For more information contact us at:
LICA
P.O. Box 158
Mill Neck, NY 11765

Thank you very much.

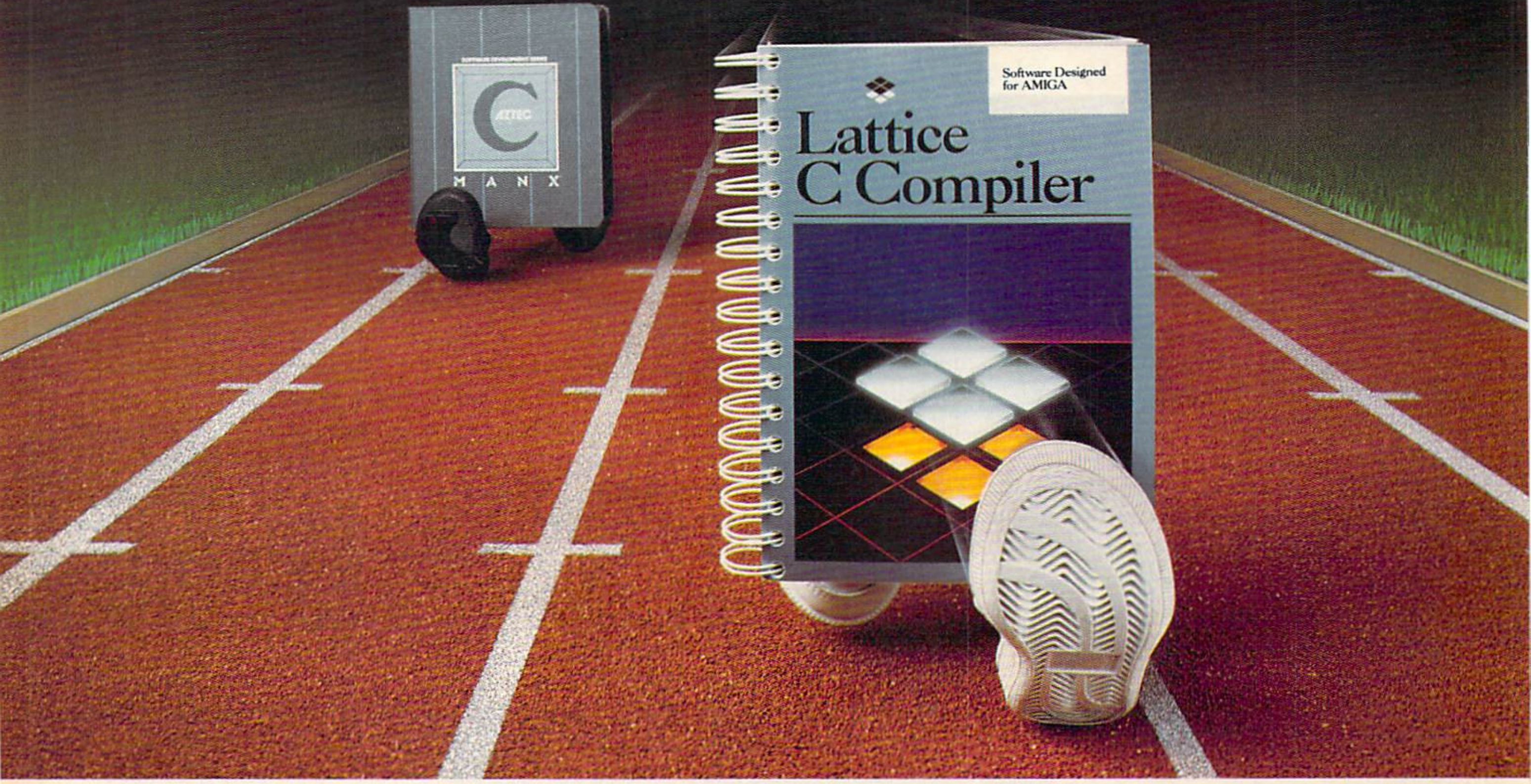
Sincerely,
Layne Dornstein
Secretary L.I.C.A.

ATTENTION ALL READERS

Do you have a suggestion or observation? Please write us. Each reader who has a printed letter with a suggestion, question, helpful technique or other useful information to help the Amiga community, will receive a certificate for five Public Domain Software disks and the gratitude of Amazing Computing and its readers.

Please send your letters to:
Dear Amazing Computing,
P.O. Box 869
Fall River, MA 02722

C who's winning the race. Lattice C for Amiga.



Lattice C has long been recognized as the best C compiler. And now our new version 4.0 for Amiga™ increases our lead past the competition even further.

Ready, set, go. The new Lattice AmigaDOS C Compiler gives you faster, more efficient code generation and support for 16 or 32-bit integers. There's direct, in-line interface to all Amiga ROM functions with parameters passed in registers. What's more, the assembler is fully compatible with Amiga assembler syntax.

More great strides. The linker, Blink, has been significantly enhanced and provides true overlay support and interactive recovery from undefined symbols. And you'll have a faster compile and link cycle with support for pre-linking.

There's no contest. Standard benchmark studies show Lattice to be the superior C language development environment. With stats like these, it's no wonder that Commodore-Amiga has selected Lattice C as the official Amiga development language.

	Lattice® Version 4.0	**Manx® Version 3.40**
Dhrystone	1294 Dhrystones/second	1010 Dhrystones/second
Float	22.20 Secs. (IEEE Format) 10.16 Secs. (FFP Format)	98.85 Secs. (IEEE Format) 17.60 Secs. (FFP Format)
Savage (IEEE)	47.67 Secs./.000000318 Accuracy	119.6 Secs./.000109 Accuracy

Going the distance. You'll experience unsurpassed power and flexibility when you choose from several cost-effective development packages. There is even a full range of supporting products, including a symbolic debugger, resource editor, utilities and specialized libraries.

You'll discover that your software purchase is backed by an excellent warranty and skilled technical support staff. You'll appreciate having access to LBBS—one of the world's first 9600 baud, 24-hour bulletin board services. And you'll be able to conference with other Lattice users through the Byte Information Exchange (BIX) network.

Cross the finish line. Order your copy of the Lattice AmigaDOS C Compiler today. We'll supply the speed. You bring the running shoes.

Lattice
Subsidiary of SAS Institute Inc.

Lattice, Incorporated
2500 S. Highland Avenue
Lombard, IL 60148
Phone: 800/533-3577
In Illinois: 312/916-1600

Lattice is a registered trademark of Lattice Incorporated. Amiga is a trademark of Commodore-Amiga, Inc.
Manx is a registered trademark of Manx Software Systems, Inc.

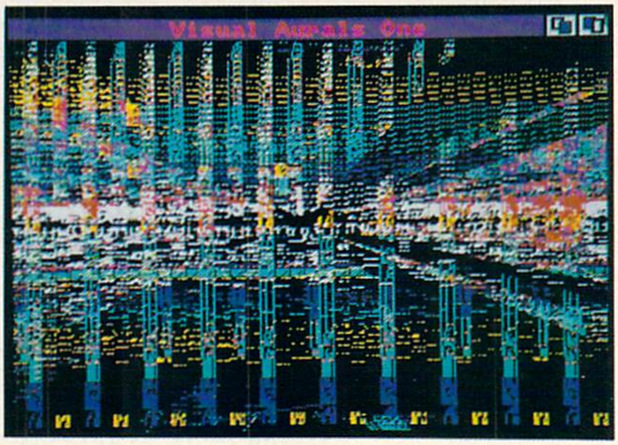
MindLight 7

The Visual Aurals Music Animation System Experience

by David N. Blank

I should begin by saying that you would have to be crazy to buy this product. It has little or no practical use. However, it is definitely one of the neatest Amiga gadgets around. If you are as crazy as I am, and a gadget freak as well, read on.

The Visual Aurals Music Animation System (VAMAS for short) by Visual Aural Animation is probably the most sophisticated personal computer-based color organ known to mankind. Before the advent of computers, a color organ was simply a device that took sound input and blinked lights in response to certain frequencies. In the past, boxes that blinked in time with music were hits at parties and discos. Enter the Amiga, capable of amazing graphics, and the VAMAS.



The VAMAS package consists of a hardware device, a disk of software, and a thick soft-cover manual. The hardware consists of a circuit board enclosed in a round, clear-acrylic case called the MindLight 7. The MindLight 7 plugs into the second mouse port. The aesthetics of the MindLight 7 are sure to please the user. It has just the right look to invoke the curiosity of onlookers, complete with two blinking LEDs on the circuit board.

The MindLight 7 receives sound input from either the sensitive internal microphone or a 3.5" stereo plug for input from a home stereo. The sensitivity and response to the sound input can be modulated using the three thumbwheels on top of the case.

Sound is analyzed and processed in real time into four "MindLight channels" and one "Wavelength channel." The MindLight Channels are: "Bass Intensity & Treble Intensity," "Overall Intensity & Overall Frequency," "Bass Intensity & Bass Frequency," and "Overall Intensity & Treble Intensity." The Wavelength channel gives a reading of the sound input's exact frequency or wavelength.

As impressive as the MindLight looks, its software is even more amazing. The software provides an overwhelming assortment of features used to visually animate the data collected by the MindLight 7. Every component of an animation can be stored in an "Environ" for later recall. Ten Environs are stored in memory for instant access, and an unlimited number may be stored to disk. The manual breaks each animation down into four parts; I'll take the same approach in describing the software.

The first component of the VAMAS software is the "vista." A vista is the description of the backdrop for each animation, and how the backdrop will interact with the animation. The VAMAS is designed to work with an optional external video signal input from a genlock device. The vista may be changed to include a mask to determine where, in relation to the other parts of the animation, the video input will be displayed. The vista also determines the location in an animation (foreground or background) of an IFF picture.

The second part of an animation is a "visual." The manual provides an excellent definition: "A visual is a sound-responsive pattern generation scheme with distinguishable characteristics." The eighty-four visuals available may be divided into three groups. The first group is the Kolai (collage-like) visual. These visuals manipulate a single object in response to the sound input. The object can be either a pre-programmed geometric shape or an IFF brush. The second type, the Mozai (mosaic-like) visual, uses many small objects to form moving and changing patterns on the screen. I have intentionally shied away from describing any individual visuals because they really have to be seen to be understood and appreciated.

The last group of visuals could be described as the miscellaneous set. Two of these miscellaneous visuals have serious uses. The Scope visual

(continued on page 10)

LIVE!™

AT LAST!

...real-time, LIVE! video on your Amiga's screen.

- ***True Color:*** just as it comes from your video source: camera, VCR, TV, anything. Direct, moving, in your Amiga's memory...our patented technology.
- ***Fast:*** video images in black & white, 32-color, and 4,096-color HAM. See 15 new images every second in black & white, 12 in color, 4 in HAM.
- ***Save:*** moving video, play it back, use it in other programs. Unlimited stills, too.
- ***Video Effects:*** real-time mouse-controled...posterization, fades, color-keying, strobes, more.
- ***Roll Your Own:*** programmer's video library, hardware documentation, examples in C, Basic.
- ***$295.*** Immediate delivery. This is hot.

To order call toll-free anytime:
Nationwide: 800-452-4445, ext. 1156
California: 800-626-9541, ext. 1156

For more information, contact:

α²

A-Squared Distributions Inc.
6114 La Salle Avenue, Suite 326
Oakland, California 94611
415-339-0339

(continued from page 8)

gives an oscilloscope-like display of the incoming intensity and frequency levels, useful for fine-tuning the software and MindLight 7. The Atune visual displays, within a musical scale, the dominant frequency as a musical note being received by the hardware. The last miscellaneous visual is a sort of random mode, called "Evolve." In this mode, the software randomly chooses different visuals and effects in response to the music. If a certain combination strikes the eye, the user can freeze the "evolution" at any time and save the current parameters.

The next portion of the VAMAS software is the Fader. Just as the user is given exacting control over what is placed on the screen, myriad options exist to determine how visuals are removed from the screen. These are called Faders. The user may choose to have previously drawn effects fade from the screen or be removed all at once (blanked). The screen can be faded or blanked to and from the background, foreground, IFF image, or video input. An interesting variation on this theme is the Kolai fader. This fader uses the same objects that normally decorate the screen in a Kolai visual. The usual complement of options is available here to determine which portion of the visual should fade, which can produce some spectacular effects. The Kolai objects seem to act as windows to another part of the animation. For all Faders, the user controls the speed of fading and blanking. The Fader can be synchronized with a MindLight channel to fade or blank in response to the music.

If all this does not suggest a mind-boggling set of options, add to them the set of effects under the Modifier heading. As its name suggests, it modifies the current visual or fader.

The list of modifiers is long. They include: selection of Kolai and Mozai objects, the speed and selection of fader type, the colors used in an animation, software sensitivity and synchronization, and miscellaneous effects. Each one of these modifiers has a different set of options. For instance, the color modifier allows the user to dictate the colors used in an animation, the speed of color change and/or color cycling, the range to cycle, and which of the three available palettes to use. The user can also choose effects such as scrolling and the scrolling modifier, called bit splitting. Bit splitting is an effect whereby each pixel is split into two component-colored pixels which then move away from the original. There are several different ways to scroll and bit split; the user is left to choose. Speed and synchronization controls are also provided. The VAMAS also provides a modifier called the V-number (a set of variations on a visual numbered from -1,000,000 to +1,000,000). The manual provides an example of how different V-numbers can show totally different aspects of a single visual.

As we run down the options and features of the package, two other miscellaneous features should be noted. The Visual Aurals Music Animation System is capable of performing its magic using the entire range of Amiga graphic resolutions, including HAM and overscan. If you tire of the Amiga responding strictly to outside sound input, you can invoke the second interesting feature. To provide a two-way street, the software has a built-in MIDI sequencer and sync support for driving external synthesizers and drum machines connected by a MIDI interface to the Amiga. An interface is not included in the package, but any standard Amiga MIDI interface can be used. All the usual VAMAS operations run simultaneously with MIDI operations.

I have very few gripes with this package, and they are minor. The first is the location and size of the MindLight 7. In place, the MindLight 7 partially overlaps the expansion port of the Amiga 1000, which is annoying to users who have attached expanded memory via the expansion port.

My second problem lies with the manual. It is fairly complete, but a bit disorganized. This can be a big problem for someone using a new product, but in this case it is much less troublesome. Digging through the manual to locate information adds to the adventure of exploring the seemingly endless options. My last caveat: the sophistication of this package has made it a bit high-priced for any but the most devoted gadget freaks.

I think the term "adventure" is the key. At first, the number of options seems to reek of overkill and can be a bit overwhelming. This impression is toned down by the sense of adventure and exploration the manual instills. It begs for experimentation; there is no wrong set of keys to push or wrong option to invoke. At worst, the user is faced with a blank screen, for which the manual offers several remedies.

There is also a feeling that all the options a user could possibly want have been included; it just may take some experimentation to produce the desired effects. The Evolve visual also helps to suggest new and different combinations every time it is invoked. The end product of this plethora of options is a lasting interest value seldom found in recreational computer packages. I am also told that Visual Aural Animation is hard at work designing a developer's kit to allow programmers to design their own applications and animations based on input from the MindLight 7. A Visual Aurals 3-D package is being designed under the direction of the author of popular public domain 3-D game Triclops. These two products should be released in the months to come.

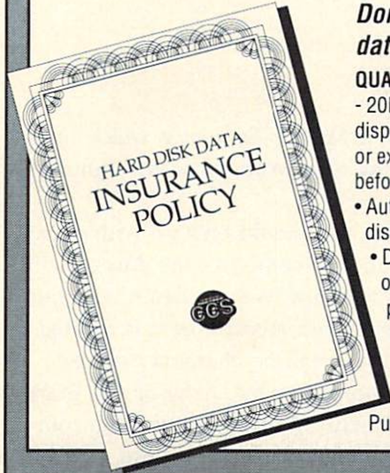
The Visual Aurals Music Animation System seems to have a practical value approaching nil. But it is definitely one of the most unusual and entertaining Amiga gadgets around.

•AC•

MindLight 7
List Price $169
Visual Aural Animation
P.O. Box 4898
Arcata, CA 95521
(707) 822-4800

NEW for the AMIGA

QUARTERBACK™


HARD DISK DATA
INSURANCE
POLICY

CCS

Don't fumble around with your Amiga files. Let QUARTERBACK manage your valuable data. The Quarterback sneak scores every time!

QUARTERBACK is a *FAST* Hard Disk to Floppy Backup Utility for the Commodore Amiga, featuring: • Fast backup - 20MB in less than 40 minutes • Uses two floppy drives for backup with automatic switching • Builds, sorts, and displays catalog of files and subdirectories • Provides Full/Subdirectory/Individual file backup/restore • Includes or excludes files by name (with wild cards), file date, or archive bit • Calculates the number of floppies you'll need before you start • Handles files of unlimited length, unlimited subdirectories and unlimited files per subdirectory • Automatically formats diskettes with no delay as it writes • Sequentially numbers and date/time stamps backup diskettes • Checks the sequence number and date/time stamp of each diskette before restoring files from it • Detects bad disks during backup or restore • Restores original date/time stamp, file notes, and protection bits on both files and subdirectories • Runs from Workbench or CLI • Produces backup/restore report to disk or printer • Beeps for floppy change • Accepts CLI parameters and batch command files • Convenient/user friendly error recovery • Multi-tasking • No copy protection • Works with all AmigaDOS compatible hard disk drives.

You'll have fewer "time-outs" with QUARTERBACK managing your file backups.

Put **Quarterback** on your team for only $69.95 plus $3.00 for shipping and handling, CA residents add 6% sales tax.

DISK 2 DISK™

Convert C64/C128 Files to the Amiga!

DISK-2-DISK™ makes it easy and convenient to transfer C64/C128 files to and from the Amiga! DISK-2-DISK programs the Amiga model 1020 external 5.25" disk drive to read and write 1541/4040 and 1570/1571 disk formats including 1541 "flippies".

• Converts Commodore/PET ASCII to AmigaDOS standard ASCII and vice versa • Transfers word processing text files (such as PaperClip, SpeedScript and Pocket Writer) to and from the Amiga for use with popular Amiga word processors • Includes 3 public domain programs for converting C64 Koala, PrintShop and Doodle files to IFF format • Finds and flags dialect differences between Commodore Basic and Amiga Basic files • Provides VALIDATE BAM and CHECK DISK utilities (VALIDATE BAM verifies the directory structure of the 1541/1571 diskette; CHECK DISK reads every block of a 1541/1571 diskette to detect diskette errors).

DISK-2-DISK requires the Amiga model 1020 5.25" disk drive.

Only $49.95
plus $3.00 shipping and handling
CA residents add 6% sales tax.

DOS 2 DOS™

Read/Write MS-DOS and Atari ST Disks on your Amiga

DOS-2-DOS™ Transfers MS-DOS and Atari ST Files To and From AmigaDOS!

• Supports single and double sided 5.25" as well as 3.5" 720KB MS-DOS diskettes • Reads/Writes 3.5" Atari ST diskettes (GEM format) • Converts ASCII file line-ending characters and provides Wordstar compatibility • Supports full directory path names, with wild cards in the file names • Allows selection of MS-DOS and AmigaDOS subdirectory and displays sorted directory listing • Formats 3.5" and 5.25" MS-DOS diskettes • Provides duplicate file name detection with query/replace options • Provides TYPE and DELETE commands • Permits renaming of files where file name restrictions occur • Remains resident to permit AmigaDOS disk swapping.

Only $55.00
plus $3.00 shipping and handling
CA residents add 6% sales tax.

CCS™

Central Coast Software™

268 Bowie Drive, Los Osos, CA 93402 • Telephone (805) 528-4906 • FAX (805) 541-4745
Dealer Inquires Welcome

MasterCard VISA

New Cli Window

1>

Interactive Startup-Sequence

by Udo Pernisz

There are two basic environments from which the Amiga can be operated, and they divide the world of Amiga users into two camps. From the Workbench, the Amiga is operated by the mouse. This popular operation mode selects from the icons, menus, and gadgets of the Intuition interface. The Command Line Interface provides an alternative environment which affords much more flexibility and power in handling files and functions on the Amiga. The CLI is available under AmigaDOS, the Amiga disk operating system. Here you type commands from the keyboard to the screen where they appear at the prompt, usually as a number followed by the angular bracket (1>). In the commands to follow, this prompt will not be repeated with the examples for command lines since the prompt is provided by AmigaDOS. The commands and functions AmigaDOS understands are described in the AmigaDOS User's Manual.

After the Workbench disk is inserted, and the Amiga wakes up, the screen displays an AmigaDOS window. After a few seconds, this window closes, and the familiar Workbench appears. While the AmigaDOS window is open, the Amiga executes DOS commands from a file called Startup-Sequence found in the :S directory of the Workbench disk. The Startup-Sequence is a command file containing a list of instructions for AmigaDOS to execute. They are executed in order, just as if they were typed from the keyboard. The Amiga is designed so commands in the Startup-Sequence file are automatically executed when the computer is restarted or reset with the CTRL/LeftAmigaKey/RightAmigaKey combination.

In its original form, the Startup-Sequence executes the instruction lines LOADWB and ENDCLI. The first instruction is a program that opens the Workbench environment. The second is a DOS command which closes the CLI window. You can modify the Startup-Sequence by adding commands such as "ECHO <text>" and "RUN Clock." "ECHO <text>" displays <text> on the screen, while "RUN Clock" displays the clock, normally invoked by clicking on its icon from the Workbench. Depending on your particular applications or various work environments you need, you may have many different Startup-Sequences on various program disks. (For example, you might have a Startup-Sequence which brings up the Amiga with a game or a word processor running.)

When using the CLI mode, you could end up with elaborate command files to automatically configure the Amiga with utilities and functions. There are two problems with this approach: (1) Once a customized environment is created by a specific Startup-Sequence, it can be changed only by typing in additional commands, or by changing the Startup-Sequence (e.g. with a line editor) and rebooting the computer. (2) An all-encompassing startup-sequence may take a long time to complete because AmigaDOS commands are disk resident (i.e. all commands are actually small programs present in the :C directory and each command is loaded into memory before being executed).

In most cases, you need only the basic CLI or the Workbench. It would be convenient to have a startup process flexible enough to allow you to choose interactively between various options by typing a suitable response from the keyboard.

Of course, you could put the Amiga into the CLI mode, then perform all the desired functions from the keyboard by typing in suitable DOS commands. However, this method requires a great deal of typing every time you start up the machine. As an alternative, you can execute a batch file.

A batch file is simply a text file containing a sequence of AmigaDOS commands. Typing EXECUTE along with the name of a batch file causes the Amiga to execute each command in the file as if you had typed the commands from the keyboard directly into a CLI window. You could customize your environment by executing a series of appropriate batch files. The only problem here is, you have to remember the names and functions of all your batch files. This approach still requires a fair amount of typing.

There is a solution, though. You need a way for a batch file to display a list of options, thus allowing the user to choose from these options with just a few keystrokes. This is indeed possible with an EXECUTE command that performs what is called a parameter substitution. Essentially, this means AmigaDOS uses what you type in from the keyboard in place of a parameter specified in a batch file. This feature is explained, with many examples, in the AmigaDOS Manual. Here is a brief review. The format of the EXECUTE command is:

```
EXECUTE CommandFile Argument
```

Argument is passed on to the command file aptly named CommandFile. Argument can then be tested and acted upon by the other command sequence control statements available in AmigaDOS. (A discussion of the powerful command files that can be configured for the Amiga appeared recently in *Amazing Computing* V3.1, p.59.)

Since the Startup Sequence is executed automatically before the keyboard is active, a problem arises when applying this feature to the special batch file named Startup-Sequence. It is impossible to type in the whole line as required by the command format.

Is there another way of utilizing parameter substitution? The concept of an interactive Startup-Sequence hinges on the answer to these questions: How can a parameter pass a value from the keyboard to an executing command file? Can the EXECUTE command turn to the keyboard for Argument input during execution of a batch file?

Yes it can, through a feature associated with using the question mark as an argument to an interactive CLI command. (This is not the question mark in a pattern such as "#?" used with the LIST command.) An example using the "?" is the DIR command. (See the AmigaDOS Manual.) If you type "DIR" from the CLI, you get a sorted listing of directories and files in the current directory. This command also takes arguments, including the filename and several other options. But when you type "DIR ?" you get a display of all possible responses to this command in the form of a command template. When followed by a colon, the display indicates the Amiga is waiting for one of the possible responses to be input from the keyboard. So, although the command line was terminated by pressing the RETURN key after the question mark, the CLI has activated the keyboard for entering a parameter value. (The LIST and DATE commands are similar to DIR.)

The important point here is that the "?", as an argument, invokes the keyboard for input to the process called by the command. It doesn't matter whether the command was issued by the keyboard (as in the example above), from another command file, or by EXECUTEing another command file (either from the keyboard or from a command file).

Therefore, if a command file contains a line of the form

```
EXECUTE FileName ?
```

(where FileName is another command file set up for parameter substitution), the screen will, by making its first line

```
.KEY Parameter_Name),
```

the screen will display

```
Parameter_Name:
```

when the line with the "?" is being executed.

It then waits for the input of its value from the keyboard. Let's apply this example to a simple Startup-Sequence that allows you to choose between two basic modes: the Workbench and the CLI. All files are put into the boot disk's :S directory (although files in the root directory may also be used, see later examples). Since a file by this name in the :S directory will be automatically executed when the Amiga boots, the name of the principal file (see Listing 1) must be Startup-Sequence. This file executes a second file, called OptSys (see Listing 2).

You can modify an existing Startup-Sequence in the :S directory using ED. ED is available as a :C directory command on the Workbench disk. Before you try the files listed here, make a copy of your original file if you want to restore the previous state. To create the file OptSys, type

```
ED OptSys
```

from the CLI and enter Listing 2.

Upon booting, the Startup-Sequence first displays the option list (five lines of ECHOing) and then executes file OptSys. The "?" causes the input template (the variable name) to be displayed. Thus,

```
Select:
```

appears on the screen beneath the line with the responses to the two options (the ":" was supplied by the operating system). This particular variable name was chosen so that it functions as a written prompt for keyboard input. You could call the variable VarNam1, but it makes more sense to give it a meaning associated with the file's function.

There are many ways to process the keyboard entry in the command file OptSys. For instance, you could first test for the empty string (RETURN key only) with the command line

```
IF "<Select>" EQ ""
```

where the quotes around "<Select>" are essential. There are also ways to define a default within the command file (see the AmigaDOS Manual).

(continued)

For more complex option structures, the first file (OptSys here) may in turn, call other files and also pass arguments on to those. You may use more than one argument. The format of the .KEY command allows an argument list with several arguments. All these may be entered, separated from each other by commas (not spaces) in the very first option request. The responses to a variable list are separated by single spaces (not commas), however.

The next example shows a possible file arrangement that makes use of this feature. The modified Startup-Sequence (Listing 3) reflects the larger number of options. The first file, now called OptSys1 (Listing 4), expects two parameters, Select and Option. Both are optional, so you can answer with a single RETURN key press. Incidentally, special conditions familiar from the command templates can also be applied: adding the suffix /A to a parameter name in the .KEY statement makes it a parameter for which a value must be supplied.

You select the default by pressing the RETURN key. This gives the basic CLI, since no other commands are invoked. OptSys1 will be the only file executed.

The second file uses a public domain utility (SHELL by Matthew Dillon) to enhance the CLI and add convenient features, such as command memory, command line editing, and function key definitions. SHELL is called up from OptSys2 (see Listing 5). When you select it, you can also opt for entering the date and time during startup. Another option available when you select the Workbench mode (Intuition interface) is a small auxiliary CLI window in the lower right-hand corner of the screen. In this window, you can do file handling from the keyboard, while using the icons for program selection with the mouse.

The reason for processing the options in more than one file is that you can design a system of command files with a logical structure resembling a one-sided decision tree. In this way, options requiring only a few commands are executed directly, while lengthy options are dealt with by executing another command file. Thus, the file OptSys2 in Listing 5 is only loaded if more than just the bare CLI is desired. This operation saves time, as a command file that uses the .KEY statement is copied into the :T directory before being executed. OptSys2 also could be further divided. Still more speed is gained when disk access is further reduced. This can be done by copying frequently needed files into a RAM: disk, and by providing for buffer space with the ADDBUFFERS command available on Workbench V1.2.

The programs given in Listings 1 through 5 are not optimized in any way; they just provide examples illustrating the two concepts that make the Interactive Startup-Sequence possible. This can be combined with the command language available under the CLI to set up a sophisticated program structure with which a wide variety of systems configurations can be controlled.

I'd like to thank John Kennan for clarifying discussions, and for suggesting that I write this article.

Listing One
Filename: Startup-Sequence

```
  ECHO "Choose options from keyboard when prompt appears."
  ECHO "For the default (CLI) press RETURN key."
  ECHO ""
  ECHO "                CLI        WorkBench "
  ECHO "            [ return ]         W      "
  EXECUTE OptSys ?  ; the "?" causes the parameter in file OptSys
;                     to be displayed, see the .KEY statement
```

Listing Two
Filename: OptSys

```
  .KEY Select                 ; defines parameter name
  IF "<Select>" EQ "W"        ; checks parameter input
    LoadWB
    ENDCLI > NIL:
  ELSE
    ECHO "Bare CLI"
  ENDIF
```

Listing Three
Filename: Startup-Sequence

```
  ECHO "Choose options from keyboard when prompt appears"
  ECHO ""
  ECHO "Bare CLI   CLI with           WorkBench  WorkBench with"
  ECHO "            Shell ...and Date              CLI window"
  ECHO ""
  ECHO "[ return ]     S        S,D         W           W,C"

  EXECUTE OptSys1 ?   ; the "?" causes the parameter names in
;                       file OptSys1 to be displayed, see .KEY
```

Listing Four
Filename: OptSys1

```
  .KEY Select,Option                  ; defines two parameters
  IF "<Select>" EQ ""                 ; checking for empty string
    ECHO "Bare CLI"
  ELSE
    EXECUTE OptSys2 <Select> <Option> ; supplies parameter values
  ENDIF                               ; to second file OptSys2
```

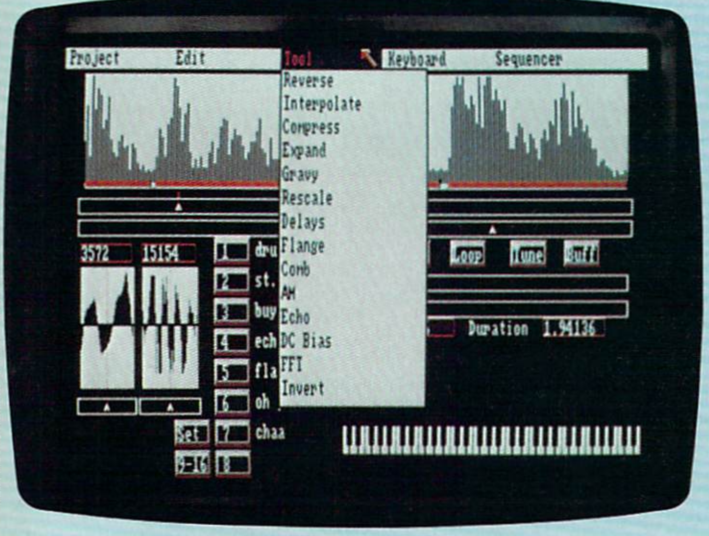
Listing Five
Filename: OptSys2

```
  .KEY Select,Option
  IF "<Select>" EQ "S"       ; response to selection SHELL
     IF "<Option>" EQ "D"    ; first check for date and time
entry
      ECHO "Enter date and time  -  dd/mon/yr hh:mn:sc "
      DATE > NIL: ?
    ENDIF
  Shell                      ; a keyboard interface program
  SKIP LastLine              ; jumps to end of file
  ENDIF


  IF "<Select>" EQ "W"       ; response to selection WorkBench
    LoadWB                   ; loads the Workbench environment
    IF "<Option>" EQ "C"     ; check for the CLI option
      NEWCLI CON:450/120/190/80/AuxCLI    ; puts up a CLI window
    ENDIF                                 ; named AuxCLI
    ENDCLI > NIL:            ; disposes of current CLI
  ENDIF

  ECHO "Selection <Select> not available" ; response to undefined

                                          ; entries
  LAB LastLine
```

•AC•

Real time video and music.

SunRize introduces PERFECT VISION and STUDIO MAGIC to enhance the look and the sound of your Amiga.



Actual Unretouched Photos

STUDIO MAGIC is the ultimate music and sound workshop with features superior to editors selling for thousands more. Compatible with model 1000, 500 and 2000 Amigas. Input sounds from stereo, VCR or microphone (with PERFECT SOUND interface) or a keyboard (with MIDI interface).

Create flanges, delays, echoes, compression and expansion to speed up (or slow down) without pitch change, do backward masking, etc. Other features include comb filter, DC bias, AM, Fast Fourier Transforms and visual representation of the sounds in the buffer.

A dozen menu driven tools allow you to make a child's voice sound like a titan or turn a TV pitchman into an alien from another galaxy. Record MIDI input in real time. Mix voices and instruments and control their playback using a MIDI keyboard. Overdub from 16 digitized "sections" and play back four of them at a time. Supports advanced MIDI features such as tempo adjust and external sync. Store sounds in IFF "instrument" or "one-shot" (8SVX) files for use with other compatible programs.

SUGGESTED RETAIL PRICE: $99.95

PERFECT VISION is the state of the art, real time video digitizer for use with model 1000, 500 and 2000 Amigas. Input from a color (or black and white) video camera or a VCR. Perfect Vision will digitize the image, display it in 4096 colors (camera input only), then store it as IFF for later use in compatible programs. Captures B&W images in 1/60 of a second—600 times faster than the competition. Color images require a bit more time. Supports 320x200 and 320x400 HAM and 16-color modes.

SUGGESTED RETAIL PRICE: $219.95

Available from Amiga dealers across America. For product information and support call:

SI *SunRize Industries*
3801 Old College Road
Bryan, TX 77801
(409) 846-1311.

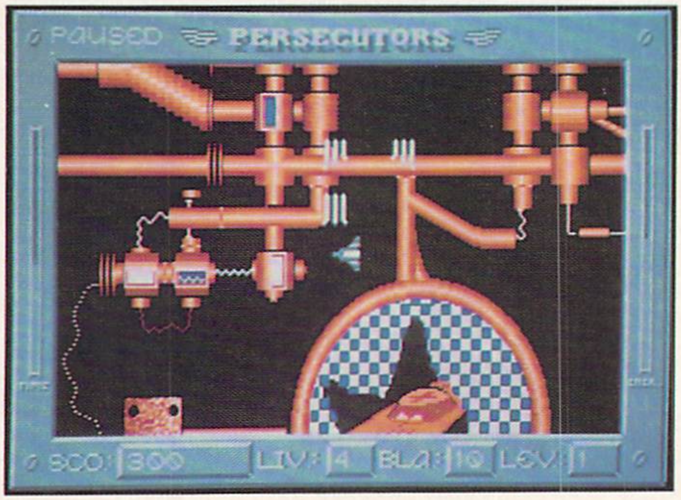
AMIGA is the registered trademark of Commodore—Amiga, Inc. Studio Magic and Perfect Vision are registered trademarks of SunRize Industries.

Take Five!

by Steve Hull
People Link: St.Ephen
GEnie: LightRaider

Roger Rabbit is the improbable star of what some sources have described as Disney Studios' most ambitious movie project in years; a quantum leap in live action/animation technology as significant in its genre as Fantasia was in its time. The movie's budget is astronomical and most individuals involved with the project have been bound with heavy nondisclosure agreements. The movie's producer? Steven Spielberg.

Needless to say, details on the movie aren't exactly crawling out of the woodwork, but this much is known: Roger Rabbit mixes live actors with cartoon characters—a lot of cartoon characters—every character in the Disney portfolio, and many characters from other studios. New special effects have been engineered to give the cartoon characters enhanced three-dimensionality.

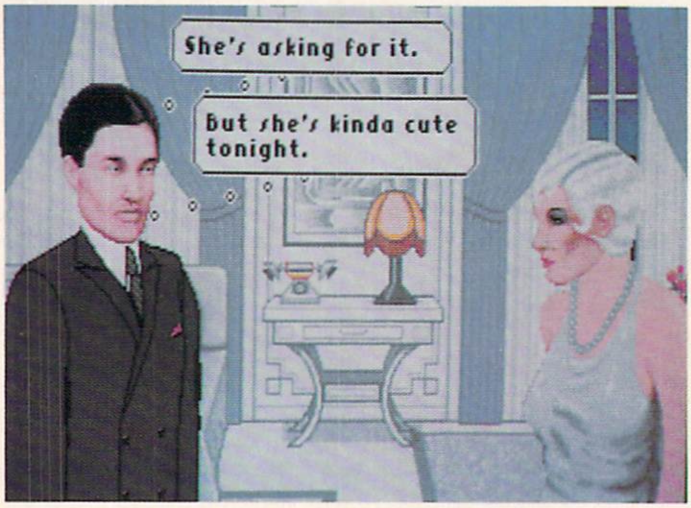


Persecutors

by Mindscape

What's all this got to do with Amiga games? Well, for months, Disney negotiators have been trying energetically to enlist a software producer to design a video game on the Roger Rabbit theme. A big budget movie, lots of special effects, Steven Spielberg—you'd think this would be the kind of project any software house would jump on, right?

Wrong. Virtually every major software producer who has been offered the project has turned it down. When investigating the reasons, the same three factors keep coming up: Disney expects the software producer to absorb the costs of production, Disney expects a large cut of the profits, and Disney wants the game done yesterday. "Disney thinks since they have such a hot property, companies should pay to produce it," one developer told me. "Companies don't understand that!"



King of Chicago

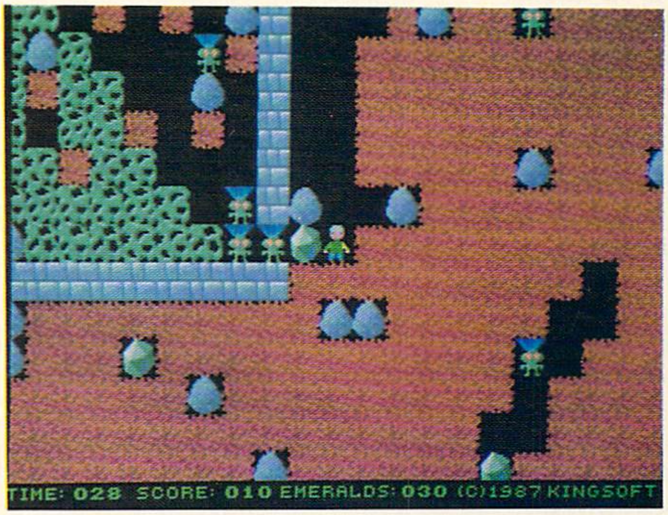
by Mindscape

Not everyone involved with the movie is gung-ho on the videogame idea. Spielberg is reportedly against the game—he feels it won't be done right.

Who is Roger Rabbit and why is he driving so many software developers crazy?

Considering the debacle over the Atari 8-bit version of ET, his concerns are understandable.

At any rate, Roger Rabbit: The Game is far from dead. Negotiations are still underway, and I may be overstating myself by saying Disney has met its match. But they have certainly found a producer who is ready to "go to the mat." Film at 11.



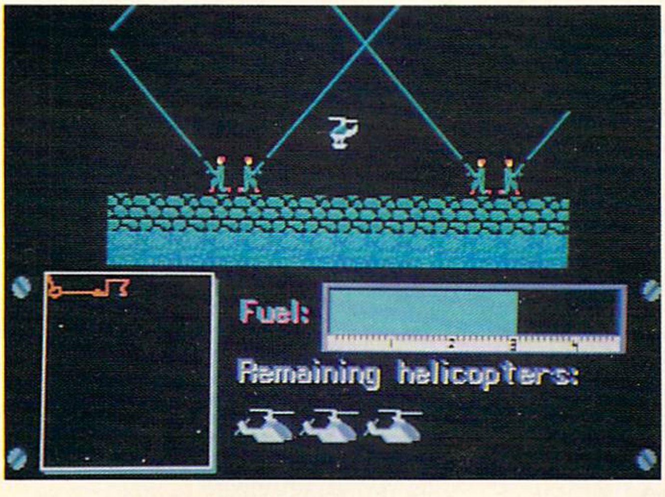
Fortress Underground

by Constellation Software

Forecast for Summer: HOT!
Accolade has promised the French import, Bubble Ghost, by the time you read this. You play the part of a spirit who must guide a fragile bubble—representing his soul—through obstacle-laden mazes by blowing the bubble around.

Rainbird Software, producer of the successful adventures The Pawn and Guild of Thieves, has unleashed yet another title on the unsuspecting world. Jinxter is an original adventure that takes place in the jinxed country of Aquitania, where you try your hand at restoring good will against the best efforts of the evil Green Witches. The graphics aren't quite up to the quality of previous Rainbird titles, but if you like British humor, Jinxter will have you on the floor.

Rainbird's long-awaited Universal Military Simulator is nearing completion, and if it's half the game they're promising, it's going to be an exciting ride. UMS allows joystick generals to recreate the greatest land-based military engagements in history on the computer screen. Included among the many scenarios are the battles of Gettysburg, Hastings, and Waterloo; each researched right down to the armaments and geography. You may reenact these battles as they actually took place, or change the scenarios—even pit armies from different conflicts against each other! Rainbird says UMS' three-dimensional gridded landscape will allow players to zoom in for views at any angle. Universal Military Simulator is scheduled for a late spring release.



Emerald Mines

by Constellation Software

(continued)

Another military simulation is due out soon from a most unexpected source—Cinemaware! Cinemaware's Bob Jacobs describes Lords of the Rising Sun as "a spiritual descendant of Defender of the Crown." Lords is a game of territorial conquest set in 12th century Japan. The map stretches across three screens. Though Lords is a war game, the playfield is not the standard matrix of hexagons and hieroglyphic symbols. Instead, game play takes place in real-time on a map "that looks as good" as the detailed map of Britain in Defender of the Crown. Storm clouds drift across the island, obscuring action; typhoons and tsunami keep things from getting stale.

As of mid-March, Cinemaware's 3 Stooges is reportedly "two bugs away" from shipping; programmers are still tweaking to get it to run on 512K, one-drive systems. I previewed this title at Cinemaware's southern California "studios;" I don't want to give anything away, but the opening sequence is worth the price of admission. Also promised for release by the time you read this: Rocket Ranger, a Nazis-from-Space pulp sci-fi story complete with a jet-pack equipped hero and the standard Cinemaware well-upholstered females. Cinemaware has written their own disk operating system for this one; it is reportedly three times faster than AmigaDOS (I know that's hard to imagine- snicker). The disk, by the way, is not copy protected—the game includes a Decoder Ring!

Later this year for Cinemaware: a "revolutionary" sports simulation, and a sci-fi title that promises to ruin your picnic in a big way.

Sometime this month Epyx expects to release Dive-Bomber, 4 x 4 Off-Road Racing, and the Amiga version of California Games. Add these titles to their recently-released Maxx-Out! game, Death Sword. The Maxx-Out! series is intended for 10-16 year old gamers, and is appropriately heavy on the action and excitement. Death Sword is a sword fighting contest set in medieval times. Your goal is to defeat successive waves of opponents and rescue The Beautiful Princess (Karateka with a sword, methinks). A real treat awaits the gladiator skilled enough to behead his opponent; a gargoyle scuttles out of the shadows and kicks the head out of the way! Yeah, the kids'll eat it up...

Psygnosis has announced the release of Obliterator for late spring. Obliterator's description sounds a little like Star Trek-Meets-Alien: a strange and sinister craft materializes in Federation space, threatening defenseless Earth. You play the part of Drak, a genetically-supercharged hero who beamed aboard the alien vessel to wreak a little good-natured havoc. I'm looking forward to this one.

Psygnosis has acknowledged that Terrorpods has trouble with the German keyboards shipped with the first Amiga 2000s, and a fix is in the works. While they're fixing the keyboard routines, programmers are planning to incorporate 68010/68020 support. The upgrade will be available free to registered owners.

Microillusions is aiming at a May release for Turbo, the third release under their One-To-One series. Not strictly the racing game some people had expected, Turbo is closer to the arcade classic Spy Hunter in its look and strategy. As has become standard fare in Microillusions' One-To-One series, Turbo may be played against the computer, or a human opponent either head-to-head or via modem.

Although the first One-To-One title, Firepower, has been quite successful since its December release, producer Reichart Von Wolfsheild is already planning an upgrade to squash a few bugs and speed up the action. A more extensive revision to the second One-To-One title, Galactic Invasion, is also planned. Good thing—Galactic Invasion's rudimentary graphics and ho-hum gameplay are a sorry disappointment after Firepower. Improvements are in the works, including better graphics and the ability to land on the planets. Work on both revisions for games is scheduled to begin as soon as Turbo is complete.

VIRUS ALERT!

It looks like Microdeal has become the latest victim of the SCA disk virus—the first shipments of the Amiga version of Time Bandit are infected. Kudos to Brian Meadows' excellent public domain virus checker, Viewboot, for finding this one. You can drop Brian a thank-you note at the Magnolia BBS, (205) 854-6407. As for your disks, the normal "cure"—running the AmigaDOS Install command—will render Time Bandit unusable. Return your master disk to Microdeal for a "sanitized" replacement!

Game of the Month

Since its inception, Cinemaware's games have tended to polarize people—some people love the movie metaphor, others can't stand it. While it's true that past Cinemaware games have tended toward flash at the expense of solid gameplay, I will let other reviewers tell you how much they hate King of Chicago. I've had a pretty good time with it.

For those unfamiliar with Cinemaware's premise, its intention is to do text and graphics-with-text adventuring one better, allowing the player to participate in a "movie" whose plot changes with the player's response to situations. In Sinbad and the Throne of the Falcon, for instance, winning the favor of the devious Libitina goes a long way toward success in your quest; anger her, and she will send an assassin after you.

The graphics and sound necessary to do this convincingly require a lot of machine overhead. There's no getting around this, and for King of Chicago, Cinemaware states the requirements on the box: 512K RAM and two drives, or a megabyte of RAM and at least one drive. Don't even try this game on a 512K, one-drive machine.

King of Chicago's increased hardware requirements result from the level of graphics and sound in the game. According to Cinemaware, over a megabyte was used for the graphics alone. The first time I played, it seemed as if my external drive accessed more-or-less constantly, slurping a constant stream of graphics, and especially sound, off the disk.

King of Chicago is set in the Windy City in the 1930's (gangsters' heyday). Capone has just gone up the river for income tax evasion (#&^$#@*! clever coppers) and the town is wide open for opportunistic young thugs like "Pinky" Callaghan. You control Pinky's actions (well, more or less control them—more on that in a minute) as he sets out to expand his domain from Chicago's northside to the southside. But you can't expect the boys from the west side or the Loop to just sit around polishing their tommy guns.

The game boasts the richest cast of characters, both in number and substance, of any Cinemaware title to date. This is almost a backhanded compliment when compared to the lush characterizations that have long been standard in such products as Infocom's text adventures, but then again, a picture done well really is worth a thousand words. It's one thing to describe a look of scornful disdain; it's another to actually see the character's eyes narrow coldly and his lip curl.

King of Chicago goes farther than its predecessors in overcoming a problem that has plagued Cinemaware titles from the beginning: lack of repeated playability. I enjoyed Defender of the Crown when it was initially released, but I'll admit the only time I'll take it out is to use it as a graphics demo for friends. If you know what you're doing, you can finish the game in about 15 minutes.

King of Chicago overcomes this weakness by introducing elements of randomness from one game to the next. The basic plot remains the same; you must wrest control from the Old Man, win over the gang, and systematically capture Chicago a region at a time. How you do this, however, varies from game to game. For example, you might play three games where pressure and intimidation is enough to cause the Old Man to pack his bags. On game #4, he'll stand up to you and before you know it, the boys will be be fitting you for cement overshoes. Changes in plot, personalities, and loyalties from game to game mean that you not only get a chance to try new strategies—to survive, you will have to.

The game has another interesting feature not found on previous Cinemaware titles. At several points the game stops as Pinky ponders two or more alternative actions, illustrated by cartoon-type thought balloons. You choose the course of the plot by clicking on one of the thought balloons, and the game continues. However, if you don't choose an alternative, or take too much time deciding, the game selects for you! Thus, the first time you play King of Chicago, you might not want to play at all—just let the game play itself in the form of a most elaborate demo!

King of Chicago has a lot of plot twists; after playing a dozen games, you're likely to run across new story elements. No matter how it works out, the game is a balancing act; you may be the Boss, but you still need to make management decisions—like how much to squeeze the speakeasies, rackets and gambling joints for income. And there are the expenses; besides your own cut, there are

(continued)

F-BASIC™

A **FAST** Compiled Enhanced **BASIC** Language

☐ The beginning programmer will appreciate the simplicity and ease of use of the F-Basic™ system. An entire disk of sample programs and a comprehensive manual are included.

☐ The advanced programmer will appreciate the many enhanced features supported by F-Basic™:

- Extensive Control Structures
- Recursive SUBROUTINEs & FUNCTIONs
- LOCAL & GLOBAL Variables
- 32, 16, And 8 Bit INTEGERs
- Ultra Fast 9 Digit Floating Point Routines
- Powerful String Manipulation Facilities
- RECORD Structures And Pointers
- Easy High Level Access To AMIGA™ Screens, Windows, Menus, Sound, Speech, And Graphics
- Direct Access To ROM Kernel Routines
- INCLUDE And APPEND Separate Files
- SNOBOL4™ Like PATTERN Matching
- Direct Access To 68000 Processor Registers and Bitwise Operations

☐ The included FastCom™ compiler is a one pass optimizing compiler generating in-line 68000 machine code. Extensive Compile and Run Time Errors. FastCom™ compiles thousands and thousands of lines per minute because it is written in F-Basic™!

A proven system.

ONLY $79.95 (U.S.)
complete
NOW SHIPPING!

Send check or money order to:
DELPHI NOETIC SYSTEMS, INC.
P.O. Box 7722
Rapid City, SD 57709
For fastest delivery call **(605) 341-2580**

F-Basic™ and FastCom™ are registered trademarks of DNS, Inc. AMIGA™ is a registered trademark of Commodore/AMIGA, Inc. SNOBOL4™ is a registered trademark of Bell Telephone Laboratories.

politicians to buy, goons to pay, and of course Lola, your semi-loyal moll and a royal pain-in-the-neck. Shortchange anyone and you may end up visiting the fish.

The game's graphics are very good, the sound effects are adequate, and the music is excellent. There are several arcade type challenges woven into the story. Some, like the gunfight with the southside gang, demand as much in the way of reflexes as any arcade (dare I say?) shoot'em-up. Other sequences, such as the blow-the-doorknob-off segment, are just flat dumb.

King of Chicago is not without its flaws. During some points in gameplay, lengthy dialogues go on between characters and all you can do is watch. This is fascinating at first, but as you might expect, it tends to get old after repeated sessions. This flaw is partially redeemed by the game's "fast forward" feature: holding down the left mouse button allows you to zip past repetitive segments.

Part of King of Chicago's challenge can be second-guessing your own character. There are occasions, for instance, when you will instruct Pinky to cool it—only to see him become more aggressive! There may be times when you find yourself wondering: hey, who's playing this game, anyway?

As noted on the box, King of Chicago contains some strong language. Much of the profanity is awkward and self-conscious; "I say let's hit that damn south-sider barbershop." While I would not go so far as to recommend Cinemaware perform a global search-and-replace with more appropriate phrases ("dagnabbit" has a certain ring to it), it should be noted that the movies after which King of Chicago is patterned did not require profanity to get the point across.

All in all, these shortcomings are more in the line of "growing pains." Nobody has tried to do what Cinemaware has done, and there are bound to be a few bumps on the way up that steep learning curve. Like the titles before it, King of Chicago is less a real movie than a "preview of coming attractions." But even there, it has far more substance than its predecessors. Five years from now, King of Chicago will likely seem as quaint and simplistic as the vintage videogame Pong seems today—but for now, it's an original and intriguing look at the shape of games to come.

Invasion of the B-Games

Did you ever notice those, shall we say, "off-brand" Amiga games tucked away on a far shelf at your dealer, and wonder if they were any good? You know the ones I'm talking about—normally imported from Europe, they come packaged in small plastic boxes slightly larger than an audio cassette, with inexpertly translated descriptions and slick cover illustrations that cannot possibly represent anything in the game itself. Though these games are priced lower than titles from most main-line companies like Mindscape and Activision, nobody likes to gamble $20 without knowing what they're going to get. So, as a public service of this column, let's take a look at four of the "second stringers" —Space Battle, Fortress Underground, Persecutors, and Emerald Mines.

Space Battle, produced by Digital Dreams, is a competent rendering of the coin-op classic, Asteroids. The object of the game is to clear an asteroid field by blasting miscellaneous space debris to bits—the challenge being that blasting big rocks produces little rocks. These smaller asteroids can be dispatched, but so may you—any impact to your fragile craft is fatal. You may maneuver anywhere on the screen in a realistic simulation of space physics; once you have established forward momentum, the only way to stop is to rotate your ship and fire your engine in the opposite direction. Pulling back on the stick sends you into hyperspace, which can get you out of a tight spot. But since it places your ship randomly on the screen, it can also put you in one.

Standard features such as solo, team, and competition modes are supported; an extra ship is awarded at for every 5,000 and 10,000 points accumulated. You are also awarded a ship for touching a quavering iridescent bubble that floats periodically by. Hostile saucers appear from time to time to fire a few ill-intentioned blasts your way. High scores are saved to the disk.

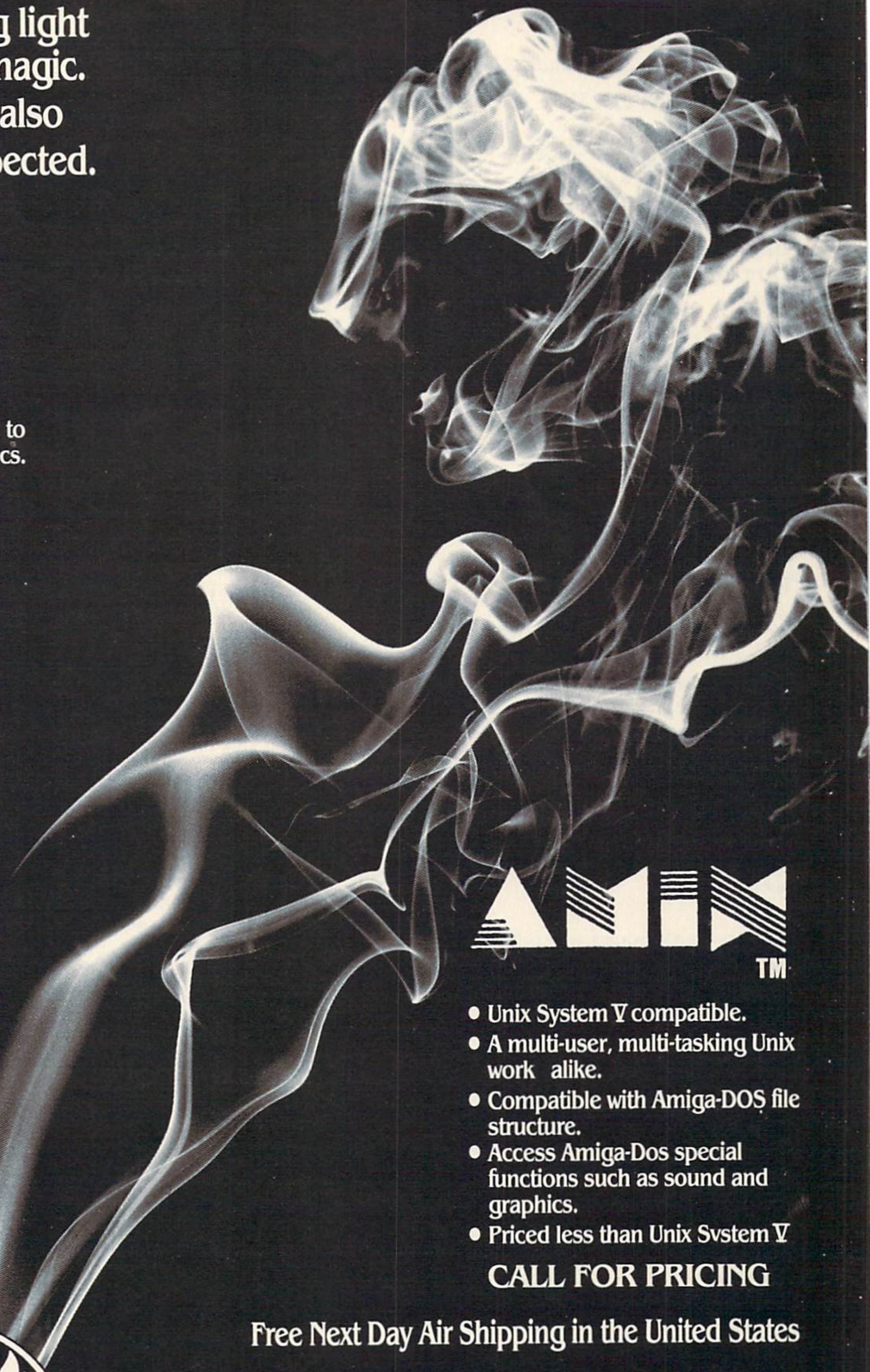
Space Battle's gameplay is very good, but it does have one annoying bug. Your ship may drift entirely off the screen for seconds at a time! While the ship has left your view, it has not left the game—and wherever it is, there are asteroids there too. When your ship floats off the top of the screen and you hear a *KABOOM* out of nowhere, you know you've bit the intergalactic dust somewhere in the twilight zone.

Fortress Underground is an engaging arcade challenge along the lines of the 8-bit game, Fort Apocalypse. It comes with very sparse documentation, but it doesn't need much—your mission is to pilot a helicopter through a huge underground cavern to destroy a hostile power station. Did I say huge? Kingsoft, the game's producer, claims it covers 640 screens. Any volunteers to test that?

Underground's graphics are functional; that's about all you can say. The first time I played, I was reminded of some of the better magazine type-in games for my old Atari 800. The sound is worse,: thoroughly unconvincing beeps and plinks that would insult any self-respecting Commodore 64.

Fortunately, its gameplay redeems Fortress Underground. You begin with seven helicopters, and must wind

(continued on page 22)



Aladdin was expecting light but what he got was magic. Our new software will also deliver more than expected.

dB PRO™

- True dBase III data file and program compatibility.
- Access Amiga® system routines, to create mixed data/sound/graphics.
- dB PRO runs applications faster than dBase III.
- Price is hundreds less than dBase III.

CALL FOR PRICING

AMIX™

- Unix System V compatible.
- A multi-user, multi-tasking Unix work alike.
- Compatible with Amiga-DOS file structure.
- Access Amiga-Dos special functions such as sound and graphics.
- Priced less than Unix Svstem V

CALL FOR PRICING

Free Next Day Air Shipping in the United States

For information or dealer inquiries:

(801) 261-8177

LAMPLIGHTER SOFTWARE, Inc.

3353 S. Main, Suite 197 • SLC, UT 84115

dB Pro and Amix are trademarks of Lamplighter Software Inc. Unix, dBase III and Amiga are not.

SYSFONT

New fonts for your CLI and workbench.

10 fonts, 5 sizes.

Fully transparent. Compatible with most text editors.

Order now! Only $24.95
Send check to:

eraware
PO Box 10832
Eugene, OR 97440

Inquiries welcome! BIX: jbaron

(continued from page 20)

your way through the cavern, while dodging the enemy's defenses. A chart in the lower-left of the screen maps your progress. You must also keep an eye on your gas gauge, and refuel when necessary. We're not talking state-of-the-art, but the game's many obstacles, enemies and puzzles are addictive enough to shave at least one point off any self-respecting high-schooler's G.P.A.

If Fortress Underground was priced at the $30-$35 level of most Amiga games, I'd think twice before recommending it. For $19.95, you'll get your money's worth.

Persecutors, a German import produced by Softgang, is the most graphically appealing of the four—and the least playable. The first challenge you will face is figuring out the object of the game. The documentation on the packaging is vague and the short blurb displayed at boot-up does not help. Phrases like, "push stick as often as possible, else the score will be decreased," have lost something in translation.

My first try at this game was fast and decisive. I went through all 15 lives without the slightest idea what had happened. "Play Again?" the game prompted. Play? I thought, "is that what I just did?"

The object of Persecutors is to navigate a small spacecraft through each of the 65 levels of play, dispatching waves of colorful alien baddies that randomly wander the screens. You must also avoid protected zones, which will vaporize you on contact. The problem is, these zones are invisible and seemingly unrelated to anything on the screen. Displays at the bottom of the screen show your score, level, remaining lives, and a number corresponding to a label which reads "BLA"—which I still haven't figured out.

Persecutors is a thoroughly incomprehensible game. Pass it by.

There comes a point in every reviewer's career when he is faced with hard decisions—Kingsoft's Emerald Mines is a case in point. Should I tell you that Emerald Mines is the biggest dog that ever wore a flea collar, and save you grief, or...

No. I must tell the truth, though some of you will curse me for it. Emerald Mines is a fine contest along the lines of Boulderdash, and like its predecessor, has caused more than its share of hapless victims to cross that fine line separating addiction from compulsion. Take it from someone who knows.

The premise of Emerald Mines is simple; you play the part of a miner whose job is to burrow through the ground in search of green and blue gems. But nothing worth having comes easily; there are a lot of ways to get yourself noisily nuked. Rattling bugs and little green antenna-waving robots wander the underground, as do menacing gray "flyers," and the "eaters"—odd little round creatures that look like Sunkist oranges with fangs. Contact with any of these is fatal, but you can avoid them, or go on the offensive by dropping boulders on their pointy little heads. Of course, that goes both ways—boulders work on your pointy little head too. That's only the beginning. Other obstacles include oozing green amoebae, acid baths, and the clock—you have only a set amount of time to complete a level. Among the tools you may use are two kinds of explosives, and keys, (if you can reach them.) One player mode and two-player teamwork is supported, and there are over 100 levels.

Graphics are only so-so, but the digitized sound effects are well-done and imaginative. As for its gameplay, this one gave me the worst case of One-More-Game syndrome I've had in months! You know the symptoms—sleepless nights, unpaid bills, indifferent spouses...Don't say you weren't warned.

I have only one gripe with Emerald Mines, and it is a relatively minor one. All of the games I have reviewed this month use disk-based copy protection, but Emerald Mines really takes things to extremes; the way my internal disk drive buzzed and vibrated, I was afraid it was going to leap out of its case! No apparent damage done, but there's got to be a better way to protect software.

Well, it looks like I've worn out my welcome for another month. Goody, get back to those games...like they say, it's a dirty job, but someone's got to do it.

•AC•

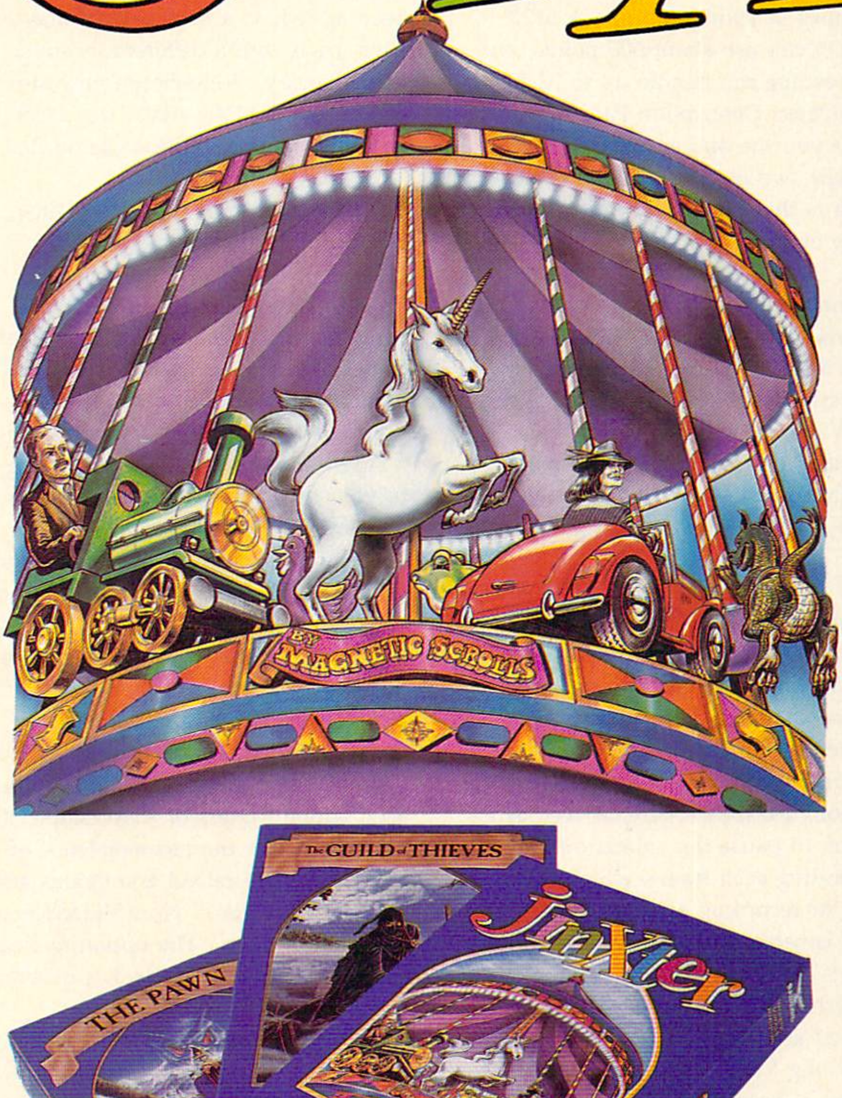
KING OF CHICAGO
List price: $49.95

Mindscape
3444 Dundee Road
Northbrook, IL 60062 (312) 480-7667

SPACE BATTLE
FORTRESS UNDERGROUND
PERSECUTORS
EMERALD MINES
List price: $19.95 each

Marketed in the U.S.
by ***Constellation Software***
17 Saint Mary's Court
Brookline, MA 02146
(617) 731-8187

JinXter



The funniest computer game since Hitch-hiker's Guide to the Galaxy.™

Enter the world of Jinxter, a challenging and hilarious graphic/text adventure from the folks who did The Pawn and Guild of Thieves. This time we decided to try something a bit different. A bit bizarre. Something to make people laugh while they tear their hair out. Sounds crazy, right? Good.

Bring a sense of humor to the task and we can promise you:

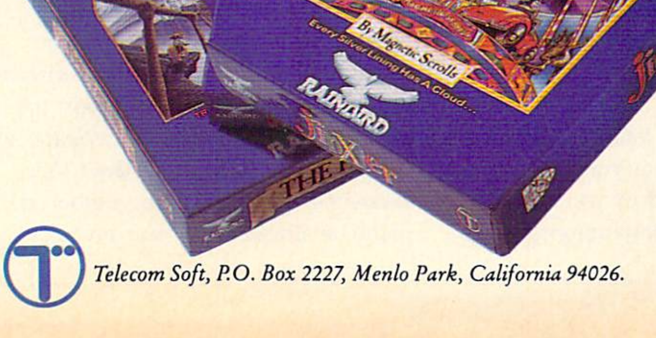
- stunning graphics.
- mind-bending puzzles and brain-tapping tests.
- a perplexing, hilarious race against time and chance.
- the state of the art in adventure game technology.
- overall game quality you rarely see anymore.

Jinxter. You'll get widdled on. And love it.

Available now on Amiga, Atari ST, IBM, C64. Coming soon on MAC, Apple II*

*Text only.

ⓉⓂ Hitch-hiker's Guide is a registered trademark of Infocom.



Magnetic Scrolls

RAINBIRD

Rainbird and Rainbird Logo are registered trademarks of British Telecommunications PLC

Atari ST© Atari Corporation.
Mac© and Apple II© Apple Computers Inc.
IBM© International Business Machine Corporation.
Amiga© and C64© Commodore Business Machine Inc.

Telecom Soft, P.O. Box 2227, Menlo Park, California 94026.

VIDEOSCAPE 3-D 2.0

by David Hopkins

The Amiga got off to a bad start. Commodore made a machine with incredible graphics and sound capabilities, but they didn't know what to do with it! There just wasn't a niche for the Amiga. Then came desktop video.

Desktop video is to the film industry what desktop publishing is to the printing industry: an easy, inexpensive way to get a job done. Animations and effects previously costing thousands of dollars can now be created for hundreds or less!

A major video product for the Amiga currently on the market is Videoscape 3-D by Aegis Development. Nothing short of incredible, VideoScape took the Amiga world by storm. True, it's not for the novice user, as the packaging clearly states, but the rewards are numerous for those willing to spend time to learn.

Now, Aegis has announced VideoScape 3-D 2.0. When Aegis called and asked if I'd be interested in a preview, I didn't know what to expect. What incredible new things did Allen Hastings have now?

After a few days of exploring 2.0, I'm even more addicted to VideoScape. Let's take a look at the complete Videoscape 2.0 system, and I'll point out the many new features.

Control Window

The Control Window is VideoScape's main screen. It consists of four "panels" and a menu bar. The Object Description Panel allows you to load an object, load a movement file, and position the object. You'll also find the number of points and polygons used in the objects you've loaded. The number of points and polygons available is limited by the amount of memory in your machine. A 512K system can use about 600 points, but Videoscape can handle up to 32,000! The Object Description Panel also gives you the option to Metamorph the last two objects loaded. The only catch is that the objects must have the same number of points.

An important new feature here is "Hierarchical Motion." It allows me to load a planet as the first object, load a spaceship as the second object, then tell VideoScape that object one is the "Parent" object. This feature makes it much easier to have the spaceship orbit as the planet moves.

Next is the Camera Motion Panel. As the name implies, this panel controls the camera. A path may be loaded for the camera to follow, and the zoom can be changed. The number of frames and tweens (groups of consecutive frames describing a camera motion) are shown here, as well as an option to pause the animation after generating each frame, which is important for recording an animation to a VCR capable of single-frame editing.

A terrific addition here is the "Track Object" selection in the load requester. By telling VideoScape which object to follow, it keeps the camera pointed at it. This feature could be used to track a speeding race car, for example.

The Viewing Options Panel lets you choose your display environment. An animation can be solid or wireframe, in any of four resolutions ranging from 352 x 220 to 704 x 440. The sky and ground colors may be selected and backgrounds and foregrounds can be loaded. Three light sources have been added, so a total of four may be used, each with a different location and intensity. VideoScape now also recognizes the PAL video standards, and adjusts its resolutions as needed.

The fourth panel is the Scene Panel. The Scene Panel is actually four buttons across the bottom of the screen. These allow you to load a complete animation setting file (object, camera motion, etc.), play your animation, save it, or quit the program. When you play the animation, a separate screen is set up according to your settings. If no camera motion was loaded, the numeric keypad is used to move the object around, or to store a single frame as an IFF picture.

VideoScape's menus have undergone the most changes. The amounts of remaining CHIP and FAST memory are now displayed constantly. In the Project Menu, a "Clear" item has also been added. Users of VideoScape 1.0 may remember the inconvenience of being forced to reload animations after playing them once. Now VideoScape holds on to them. The option to clear all settings, or just objects, is a nice one.

"Import Modeler Object" is another new item. While multi-tasking VideoScape with Aegis' upcoming Modeler 3-D, (an incredible object design tool I'm not allowed to discuss yet), this feature lets you import an object from Modeler to VideoScape so you can use it instantly. *About VideoScape 3-D* gives you the number of points available with your memory.

My 1.5 Mb 1000 says I'm allowed 1900 points. Last here is "Quit." You can probably interpret that yourself.

The newly added Display Menu lets you load your own color palette into VideoScape 3-D. It uses thirty-two colors for shading; sixty-four, if you turn on the "Extra Halfbright" selection (for those machines with the Extra Halfbright chip).

Perhaps the most important addition is "Hold and Modify," which turns on the HAM mode and lets VideoScape use all 4096 colors. The feature Aegis calls "our answer to ray-tracing" is WONDERFUL! As you might expect, it takes a long time and a bit more memory to generate a HAM frame, but the effect is wonderful. A demo showing off the new effects involves a gold-plated mask rotating over a checkerboard, with a transparent reflection on it. Each frame, one of which is shown here, took four minutes to generate on a standard 2000.

Another new menu, Extras, has a slew of new options. "Nonlinear Morph" makes the points of the morphing object accelerate at the start of the morph, then deccelerate into their new positions. "Nonlinear Zoom" does the same thing with the camera zoom function. Selecting "Black Outlines" outlines all the polygons in your object in black, defining the separations more clearly. "Crossed Dots" makes single point polygons into small plus signs—for example, the adjustment makes stars more visible in higher resolutions. "No Dithering" is mainly for use when recording the Amiga's composite video output because the dithering used in non-HAM solid scenes tends to flicker in some cases. The "Use Z Buffer" option changes the way VideoScape handles its hidden surface removal. Using this feature requires a lot of memory, as much as 1.2Mb for a 704 x 440 scene!

Finally, the Record Menu has been enhanced. As before, "Begin and End ANIM Recording" items are here, allowing you to save short animations to disk or RAM for later recording or showing on a standard VCR. The technique for saving has been improved: a VideoScape 2.0 animation takes up half as much space as a 1.0 animation. These new ANIM files can be played back with version 4.0 of ShowANIM or 4.2 of PlayANIM. ShowANIM is included on the disk. The "Begin and End IFF Save" items are new, and could come in handy. This feature saves each frame to disk as a numbered IFF picture. The pictures then can be loaded into any IFF program, such as DeluxePaint, brushed up, loaded back into VideoScape, and recorded as an ANIM file.

The last item here isn't exactly new, it's just easier to find. "Send To Serial Port" allows animations to be sent into special drivers to run professional video editing consoles.

(continued)

EGG (Easy Geometry Generator)

Included on the VideoScape disk are three programs to help you design your objects. Unfortunately, none of them are very user-friendly; they are often downright impossible! The first of these programs, the Easy Geometry Generator (EGG), is probably the most useful.

EGG presents the user with a dull text screen displaying a list of nine different "Geometry Types," including boxes, spheres, cones, and cylinders. After you select your type, you must answer a series of questions regarding the dimensions of your selection. Since meters are VideoScape's main unit of measurement, they are the ones you must use. EGG then generates an object to your specs and saves it to your disk. Most users will also find a purpose for the spherical starfield and ring of distant mountain geometry types, which are now available here.

One of EGG's main problems is that you don't get to see what it is generating. Running an object through EGG a few times before you get it the way you want it is not uncommon. Also, it seems to want to write the file only into the GEO directory on your VideoScape disk. As hard as I tried, I couldn't change directories. Luckily, Aegis added the option to abort EGG by hitting "0" at the main prompt. In the previous version, you had to enter some data, then tell it not to save before you could exit. It's not a big improvement, but it's better than nothing, I guess.

OCT (Object Composition Tool)

OCT is used to make color changes to objects, resize them, glue them together, and rotate them. Another dull text screen asks for the name of an object on your disk. After you load the object, you must answer a series of questions: if and how the object should be transformed, scaling factors, color changes, and the name of the new file. OCT also allows you to convert old format VideoScape objects into new Binary Format. While Binary Format makes objects load noticeably faster into VideoScape, they can't be edited in this format. Therefore, save your finished object as a Binary file to speed up the load time.

OCT also suffers from EGG's problem—you don't get to see what you're working on! Putting wings on an airplane, for example, is hard when you can only guess the locations!

D3D (Designer 3-D)

This program, formerly known as "ROT" (a public domain object generator by Colin French), is supposed to allow you to build your object by drawing it on the screen. As you choose each point, you must move the point and polygon number registers. This process becomes confusing very quickly. Fortunately, Aegis Modeler 3-D, which I mentioned earlier, will do all the things that EGG, OCT and D3D do, but it'll do them right!

Until Modeler becomes available, the best method is to enter the data by hand. This means designing the object on a piece of graph paper, figuring out the coordinates of each point, and then typing them into a text file. It really isn't as difficult as it might sound. In fact, Allen does most of his objects in this way. A beautiful Lotus automobile included on the disk is made of 554 points and 439 polygons!

New Colors and More

Besides the Halfbright and HAM modes, a few new colors have been added to the fixed palette. Two shades each of purple and cyan are now available, and any colors can be glossy, matte, or luminous. A new transparent feature is also available; it checkerboards the background color and the color of your object. Other new colors create bright, shadow, or chrome effects. The possibilities are amazing!

An Overview

VideoScape 3-D 2.0 is an excellent program, but like everything else, there are a couple of problems—fortunately, they aren't major ones. ANIM files made with the Extra Halfbright feature can't be played on a non-halfbright machine. Also, you can't use VideoScape with less than 1 Mb of memory. Even though it is possible to run on 512K, the program really can't operate at its full power.

Movement in VideoScape is defined by traditional X, Y, and Z (width, height, and depth), and Heading, Pitch, and Bank (rotations around X, Y, and Z axis). While these terms are certainly familiar to pilots and mathematicians, many people will find them hard to follow. Aegis included a cardboard device for the user to put together that illustrates these principles nicely, and makes learning easier.

The manual explains features well, with plenty of examples. The best way to learn, however, is to experiment. VideoScape is a lot of fun to work with once you get the hang of it, and I fully recommend it. With Allen constantly working on new features requested by users, VideoScape will always be a step ahead of the rest.

Many VideoScape users send copies of their animations to Aegis. The quality of material is impressing even Aegis, who will soon hold their second desktop video contest. My favorite was a very detailed space saga created for a child's birthday party. The program lasted several minutes and looked professional. I can hardly wait to see what people do with 2.0!

The VideoScape 3-D 2.0 package includes all the things already mentioned, plus two disks packed with objects, backgrounds, foregrounds, and sample animations. Aegis will sell 2.0 for $199.95, and it should be available by the time you read this. Registered owners of version 1.0 will be notified about an update.

•AC•

New Cli Window

1> **by Rich Falconburg**

The Command Line

A Beginner's Guide to the Amiga CLI

Getting Started

Without question, the Amiga stands alone in flexibility and power, at a price the ordinary user can afford. Unfortunately, many of the real strengths of this wonder-machine go unnoticed or under-utilized. Even experienced users often overlook some of the flexibility and power of its very capable Operating System. This column will attempt to clear the fog around AmigaDOS and its sidekick, the Command Line Interface (CLI), and demonstrate various methods to make your time at the keyboard more productive. To insure that everyone has a common knowledge base and will understand the terms and concepts I use, I'll start at the beginner level and work up.

From here we'll graduate to subjects and methods even experienced users will find helpful. I intend to cover each command in detail, grouped in a logical order of use, and investigate the various options of each. The discussion will be tutorial in nature and, for some of the more knowledgeable users, tedious at times. I will assume, however, most readers are familiar with the Workbench, and that they have have given the Introduction to Amiga at least a cursory glance.

For the tutorial's duration on the AmigaDOS, I will use the basic Workbench boot disk as shipped with the computer. All examples are shown with the input from the user underlined and, unless otherwise stated, terminated by pressing the RETURN key. AmigaDOS commands will be printed in all capital letters.

Along the way, I'll interject various methods to simplify operations and I'll make suggestions for alternative approaches. Ensuing articles will examine different aspects of AmigaDOS including but not limited to:

- Manipulating Files and File Contents
- Devices, Assignments, and Logical names
- Mountlist and Editors
- Startup-Sequence and Batch Operations
- Public Domain Commands and CLI alternatives
- AmigaDOS Replacement Project (ARP commands)
- Hard Disk Organization

The order may change, and more topics are sure to be added. This gives you some idea of the type of information that will be covered. In this installment, we'll begin the AmigaDOS Command tutorial with a discussion of the directory structure, and cover the various aspects of manipulating and navigating directories. However, the first order of business is to establish a window to the wonderful world of CLI.

Getting Started

The process for opening a CLI window using a standard Workbench disk is somewhat different between the A500/A2000 machines and the older A1000 model. Double-click on the Workbench disk icon. For the A500/A2000 users, one of the icons displayed in the Workbench window is labeled "CLI."

The A1000 users must first set a switch in Preferences. Double-click on the Preferences icon to bring up the Preferences screen. In the lower left-hand corner of the Preferences display, just above the color sliders, is a gadget labeled CLI. Place the pointer in the ON box and click. Save the preferences and exit. Double-click on the System drawer and another window will open to reveal several icons, one of which will be labeled CLI. This is our real starting point.

Now, regardless of system, double-click on the CLI icon and another window will open (Refer to Figure 1). This is known as a console window. It might be preferable to re-size the window to full screen. This kind of window is capable of displaying text only. The environment is known as the CLI. The 1> that is displayed is the CLI prompt. The number is provided by AmigaDOS to indicate which CLI you are presently using. Remember, this is a multi-tasking machine and it's possible to open as many CLI windows as memory allows. A nice thought, but it's rarely necessary to have more than two or three open at once.

The CLI environment allows us to enter commands to cause the computer to perform any number of operations. These operations vary in nature, from examining the contents of a disk, to controlling over-all system operation. At this level the user is provided direct and immediate communication with the computer.

(continued)

Some simple editing is provided in the CLI. To delete the last character entered prior to pressing the RETURN key, use the BACKSPACE key. If you need to delete the entire line, press the control key, and holding it down, press the X key. No, it's not possible to edit the line using the cursor keys; and the delete key prints only a checkered box. But all is not lost. Several enterprising programmers have written programs such as CLE, MyCLI, ConMan, and others, that help correct these deficiencies. I'll be covering these and several other Public Domain programs in future editions of this column.

The Information Shuffle

Before we get in too deep, a short description of the Amiga's disk file system is in order. The Amiga is equipped with one internal disk drive. This is a physical device referred to as DF0:. Although AmigaDOS is aware of a device called DF0, it doesn't really use that designation for directory operations. To make everything easier, AmigaDOS uses the Volume Label (disk name). This approach greatly simplifies single disk operation. Each drive is seen as a generic Input/ Output device.

By this, I mean AmigaDOS sees the physical device (DF0, DF1, DH0, etc.) as the vehicle for the information, something like an "In" or "Out" basket. This means the same device may be used by different Volumes, and AmigaDOS will keep track of each. You have probably experienced this feature several times already, especially if you have only the one disk drive. When you issue a command or otherwise attempt to access the Workbench disk without the disk in the drive, you often see a requestor asking you to "Insert Volume Workbench in any drive." "Any drive" means just that. If you pop the Workbench disk in DF3: (assuming you have that many), AmigaDOS will happily accept it and continue. (But you may have to click Retry first. The Workbench disk is unique.) The system found the disk by looking for the Volume Label. I'll explain this in greater detail when we get to assignments and logical operations.

AmigaDOS provides for a layered directory structure, sometimes called hierarchical, in which data are divided and sub-divided into organized groups (refer to the diagram in Figure 2). The top layer is called the ROOT directory. When we look at the device using the designation DF0: or the Volume Label, this is the directory we see.

The next row of directories represents the division of the ROOT directory into the organized groups mentioned, each separate from the others. Several of these are sub-divided further; this chain can continue much farther than most users will need. (For the sake of brevity, only the directory names and a few files are shown. In reality, there are a number of files contained in each directory).

To supply AmigaDOS with a beginning reference point, certain things are assumed by the system. These assumptions are called defaults, the values AmigaDOS will use until you inform it to use something different. The default directory is set to the boot volume, as in WORKBENCH 1.2:.

AmigaDOS file names are quite flexible and allow up to 30 characters. All special characters (+,-,=,_,%,space, etc.) may be used with few limitations except for the slash (/) and the colon (:). (I'll explain why later.) Using special characters requires the entire specification be placed between quotes. For example, you'll often see names with spaces in them, such as:

```
Read Me
```

For AmigaDOS to recognize this name it must be entered as:

```
"Read Me"
```

If the device or volume name is included, it must be inside the quotes as well. To use quotes ("") in the filename, precede them with an asterisk (*). The asterisk must be entered twice (**) for AmigaDOS to accept it. Now, grab your keyboard and let's take a tour.

Directories

Click inside the console window to activate it. You'll find that this particular step is easily overlooked. As with Workbench operations, the window must be active to accept input. Let's verify where we are first. This is done using the CD command with no parameters. Enter CD and press RETURN. You should see the following:

```
1> CD
Workbench 1.2:
1>
```

CD means Current Directory. Entered by itself, it will always display the directory your default is currently set to. The colon indicates that this is the ROOT directory and is appended to the Volume Label. Because the colon is used by AmigaDOS this way, it cannot be used in file names. Generally, the Volume Label and the root directory name are always one and the same. Here, I interject the first of Rich's Commandments of CLI:

Know Thy Default!

Get in the habit of checking or confirming your present default. Believe me, if you do, you may be spared the "agony of delete" from overwritten files—or worse, deleting the contents of an entire directory. With the power of AmigaDOS comes greater responsibility. Some wise person once said that "a computer is a device designed to allow humans to make an ever increasing number of mistakes and to make them faster." Having been a victim of this, I wholeheartedly agree. I hear one of the experienced users mumbling something about Public Domain commands that display the default for you all the time. Agreed, but before we get to what AmigaDOS should be, I want to cover what AmigaDOS is. Besides, not all the P. D. commands always work properly. (I use the improved PROMPT command provided by ARP, but that's a discussion for another time.)

To use CD to navigate the directory structure we need to determine what is on the disk. Do this with the command DIR. For example:

```
1> DIR
          c (dir)
          System (dir)
          l (dir)
          devs (dir)
          s (dir)
          fonts (dir)
          libs (dir)
          Utilities (dir)
.info               CLI
CLI.info            Clock
Clock.info          Disk.info
Preferences         Preferences.info
```

This, or something like it, is what the Workbench disk ROOT directory will resemble. Notice how everything is neatly separated for us. At the top of the list we see several entries followed by the (dir) notation. This indicates a sub-directory. In this case, the directories listed are sub-directories of the ROOT directory. By "sub" I mean it's located "under" the current directory.

Let's look at the contents of the devs directory. There are two ways to do this. If we don't want to change our default, it's possible to provide the DIR command with the name of the directory. For example:

```
1> DIR devs
          keymaps (dir)
          printers (dir)
          clipboards (dir)
clipboard.device              mountlist
narrator.device               parallel.device
printer.device                serial.device
system-configuration
```

This operation can be taken even farther. To look at a specific sub-directory, all you have to do is separate the names with a slash (/). Now you see why that character cannot be used in filenames. Enter the following to look at the printers directory:

```
1> DIR devs/printers
Alphapro_101        Brother_HR-15XL
CBM_MPS1000         Epson
Epson_JX-80         generic
HP_LaserJet         HP_LaserJet_PLUS
Imagewriterll       Okidata_292
Okidata_92          Okimate_20
SG10                StarNX10
```

This directory "stacking" can be used with a number of commands. The term used to describe this "directory/directory/directory..." list is path. It's also the command PATH, used by AmigaDOS to define a specific directory structure. Every time you enter a command, AmigaDOS first looks in the current default directory and then in the C directory for a file by that name. If it's not found, the message "Unknown command" is returned. By using the PATH command, you may define other directory structures for AmigaDOS to search. For example, the standard startup-sequence file establishes the paths for finding the SAY, DiskCopy, Notepad, Calculator and other commands.

An Easier Way

Entering the entire path specification every time we need to do something in a directory soon becomes monotonous. Fortunately, the CD command makes life easier. For example, let's change our default to the printers directory listed above like this:

```
1> CD devs/printers
1> CD
Workbench 1.2:devs/printers
1>
```

The second CD was used to confirm our new default. Now just enter DIR and you will see a listing the same as above. Figure 3 may help clear up what we just did. As shown, we moved down the path to reside in the printer's sub-directory. Now, to see the contents of the ROOT directory, we must use a path specification. There are several ways to do this (each of the following is equivalent):

```
1> DIR df0:          <—absolute (device)
1> DIR :             <—absolute (for ROOT
                          of default Volume)
1> DIR //            <—relative
```

As you can see, the DIR command is quite flexible. Each of these examples will produce the same listing. Absolute means to define the complete path specification, starting at the top of the hierarchy. The relative specification may be new to you. It tells the system you are using the current default directory as a reference and that you want to display the directory that is two steps above it. If you said, "I'll bet each of those methods could also be used with the CD command," give yourself a cookie and go to the head of the class. It's often convenient to jump from one sub-directory to another without entering the entire path. For example, to CD to the keymaps directory from the printers directory simply enter:

```
1> CD /keymaps
1> CD
Workbench 1.2:devs/keymaps
```

You're not forced to specify the path from the top down every time. I know it doesn't seem significant now, but if you ever get a Hard Disk you'll find this feature very useful.

Using the Options

There are times when it would be nice to see the contents of every directory on the disk. The DIR command may be used with several options to modify the way it displays information. The first one is "opt a," used as shown:

```
1> DIR opt a
```

(continued)

DATASOUND
"MINDS OVER MIDI"

SYNTH WIZARD Dx7/Dx7 II Editor Librarian $99.95
SONIC SPECTRUM SAMPLE LIBRARIES
Rock & Roll .... $59.95 Special F/X $59.95
Classical .... $59.95 Percussion $59.95

Each 5 disk group contains 60
Professionally sampled sounds that can be used with
all currently available IFF compatible music software.
Check or Money Order:
VA Residents Add 4.5% tax, Add $2.50 for P/H
Allow 4 - 6 weeks delivery.
DATASOUND • 603 Brantley • VA Beach, VA 23452 (804) 431-1362

I'm not going to show the output of this one. Try it, you'll see that it displays all directories and their contents. The indentation indicates a sub-directory; the files shown directly under it belong to that sub-directory. The exception is the ROOT sub-directories. Their non-directory contents are displayed after the last sub-directory and before the next directory in the ROOT directory (whew!). This option may also be used with any path specification. If you want to see only the directory names use:

```
1> DIR opt d        or
1> DIR opt ad       <—show all directory
                       names
```

There is another way to look at each directory on the disk and skip around without using the separate commands of DIR and CD. This is the "opt i" or interactive option. The following commands may be used after you enter DIR opt i and the question mark is displayed.

e	"enters" into that directory
RETURN	press RETURN to go to the next directory
b	backup to previous level (stops if at ROOT)
t	TYPES or prints the file contents to the screen (control c aborts)
del	DELETES current file or directory (If empty)
?	shows template for the interactive option
q	quits interactive mode

If you attempt to "enter" a file you'll get an error message. All options may be used in combination (e.g. opt dia).

Organizing a Disk

Now that we've learned about the directory structure and how to get around in it, let's see how to use a command to set up our own directories. For instance, you soon find that saving every letter you type in one directory becomes difficult to manage. How do we organize our data files? With directories of course! AmigaDOS provides us with a convenient command for creating our own directories. Guess which one? If you said MAKEDIR, pat yourself on the back. Here's an example of how we use it.

It would be nice to keep the letters we send to our friends and relatives separate from official correspondence. The data disk has been formatted with the name Letters. We'll create two directories on our data disk with the names Family and Business. Here's how:

```
1> MAKEDIR Letters:Family
1> MAKEDIR Letters:Business
```

Yes, you could substitute the Volume Label with the device name such as:

```
1> MAKEDIR DF0:Family     or
1> MAKEDIR DF1:Family
```

If you have only a single drive system, the first example will reduce disk swapping. If you check the directory of volume Letters, you'll see the two new directories created. You could also CD to the volume or directory you want to create a new directory in, then enter MAKEDIR and the directory name. Now, when using your text editor or word processor, specify the directory by preceding the file name with the name of a directory and the separating slash. For example:

```
Letters:Family/Tommy
```

(The Volume is unnecessary if that is your default.)

I hope this introduction to AmigaDOS directories has helped you. Next issue, we'll learn how to use the LIST command to display various attributes about our files, as well as several commands used to manipulate files and directories.

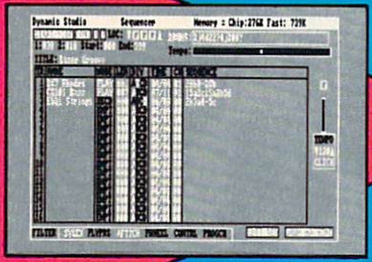
•AC•

Address all correspondence to the author in care of this magazine.:

Amazing Computing
P.O. Box 869
Fall River, MA 02720

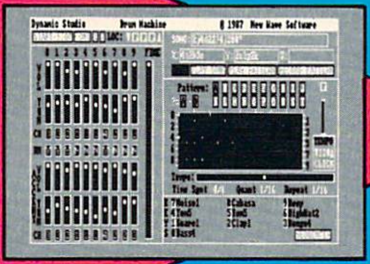
DYNAMIC STUDIO

Integrated Music Software for AMIGA computers



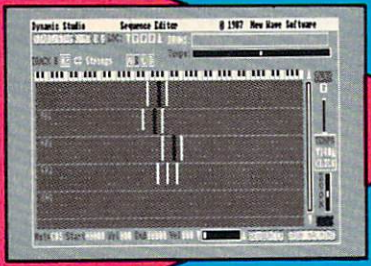
Sequencer

- 16 Tracks
- 64 Individual Sequences
- ACCURATE timing 1/192 note resolution
- User definable time signature for each track
- Extensive Track Manipulation Functions
  - Quantize (3 methods)
  - Controller Scaling
  - Echo
  - Time Shift
  - Merge
  - Randomizing Functions
- Built-In System-Exclusive Librarian
- SMUS compatable (Score with Deluxe Music ™)



Drum Machine

- Eliminates the need for costly external drum machine (uses the Amiga's internal sound channels)
- 200 digital drum samples included or use any IFF (one-shot format) standard file. All Amiga samplers support this standard
- Graphic Editing of Drum Patterns
- Adjustable tuning and volume ranges for each drum
- Velocity Sensitive (using external MIDI keyboard)



Event Editor

- Text Editing
  - Translates MIDI data into easy to understand statements
  - Modify, insert or delete any type of MIDI data
- Graphic Editing
  - Insert, delete and edit notes visually
  - Displays track in "Piano Roll" graph

AVAILABLE NOW! List Price **$199.95** No Copy Protection
1 Megabyte Recommended

NEW WAVE
SOFTWARE

P.O. Box 438, St. Clair Shores, Michigan 48080, (313) 771-4465

© 1988 New Wave Software, Inc. Deluxe Music Construction Set is a Trademark of Electronic Arts Inc.

Traditional Quality and Value

Is Amazing Computing™ a little old fashioned? Each month, Amazing Computing™ provides its readers with the finest techniques, reviews, and special features for the Amiga. The AC staff has one criterion: each issue is developed, shaped, and crafted into a publication our writers want to read.

For over two years (over 24 issues) AC has provided more pages and more information on the Amiga than any other resource. Amazing Computing has supplied more in depth and consistent information to the Amiga user than any other publication.

Amazing Computing enjoys a long line of firsts. AC was the first publication to document the Amiga's Command Line Interface and the only magazine to show the Amiga users how to improve their video output. Amazing Computing was the first major publication to offer program listings and inexpensive public domain software while providing the first programming hints and techniques.

Although Amazing Computing's history is important, the future is even more exciting. As the Amiga continues to grow with new hardware and software, AC will continue to provide its readers with the most complete information available. With a past like Amazing Computing's, the future must be a commitment to quality and innovation.

Save over $23.40

As a subscriber to Amazing Computing, you are guaranteed to be informed and up to date on the rapidly growing Amiga market at **a savings of over 49%** of the newsstand price. A one year (12 issue) subscription to AC is only $24 US ($36 Canada & Mexico, $44 overseas).

Is Amazing Computing a little old fashioned? In a world of glitz and glamour, where style replaces substance, and fluff can dominate quality, giving value in exchange for value received may appear old fashioned. At Amazing Computing, we believe it just makes good sense.

☐ Yes! Please start my 12 month subscription to the original Amiga monthly resource, Amazing Computing™. I have enclosed $24.00 for 12 issues ($36 Canada and Mexico, $44 Foreign Surface). All payments are US funds drawn on a US bank.

Name ______________________________

Address ______________________________

City ________________ St ______ Zip ________

Send with a checks or money order (no cash or credit cards) to:

PiM Publications, Inc.
P.O.Box 869
Fall River, MA 02722

All rates are one year only. Foreign and US air mail rates available upon request. Please allow 6 to 8 weeks for delivery.

Hot on the Shelves!

The Latest Amazing Amiga Developments

by Michael T. Cabral

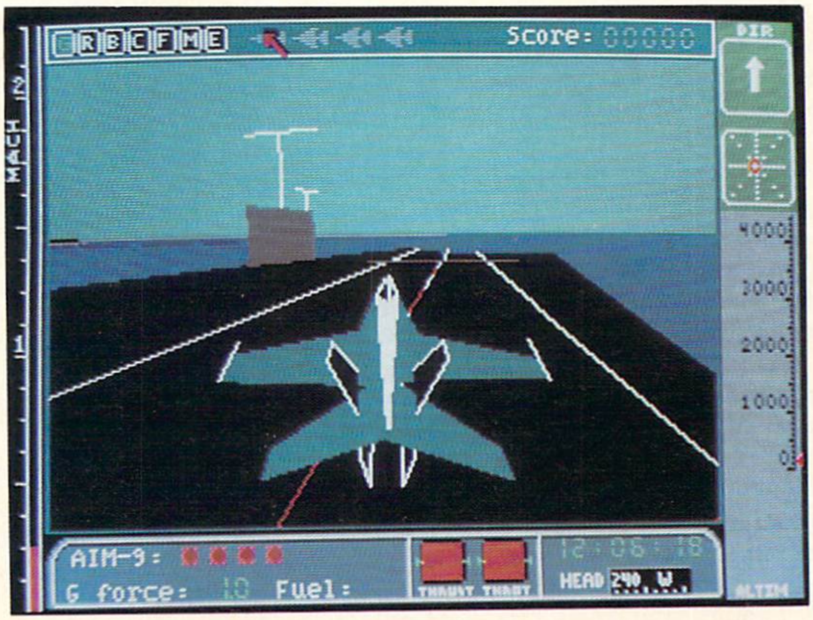
Into the Wild, Blue Yonder!

Move over, Chuck Yeager. Thanks to **Jet**, the latest from subLOGIC Corporation, Amiga owners can now take control of our armed forces' hottest flying machines. The F-16 Fighting Falcon tactical craft and the Navy's F-18 Hornet multi-role fighter are at your disposal—it's up to you to put them to work.

Jet offers a number of game options and flight features. You can hone your flying skills in the free flight mode, or head straight into the real thing: honest-to-goodness air battle. Various land/sea target strike scenarios and nasty, computer-guided MiG-21 and MiG-23 fighters await cocky Amiga aces. A full-screen "Heads-Up" display of both your instrument panel and your environment gives you a fighting chance. Also on your side, you've got search radar, automatic target tracking, an arsenal of combat ordnance, and an ejection seat ... just in case.

Jet also gives you a number of different vantage points. 3-D windows let you eye the scene from the cockpit, spot plane, or tower. If you really want to follow through on your actions, a new feature lets you follow your missiles right smack to the point of—*Kaboom!*—impact. If you prefer a safer outlook, check the full screen or map viewpoints.

Like its predecessor, subLOGIC's Flight Simulator, Jet allows you to take personal battles to the air. A serial port hook-up to a second Amiga lets you battle a fellow pilot head-to-head until one of you goes tail-spinning into oblivion. For the more relaxed moments, there is plenty of scenery to take in, including support of the subLOGIC Scenery Disks. So, if you think you can cut it as a fighter pilot, or even if you just want to go out for a Sunday fly, jump into the cockpit of subLOGIC's Jet.



Jet from SubLogic

Jet
$49.95

subLOGIC Corporation
713 Edgebrook Drive
Champaign, IL 61820
(800) 637-4983 (order line)

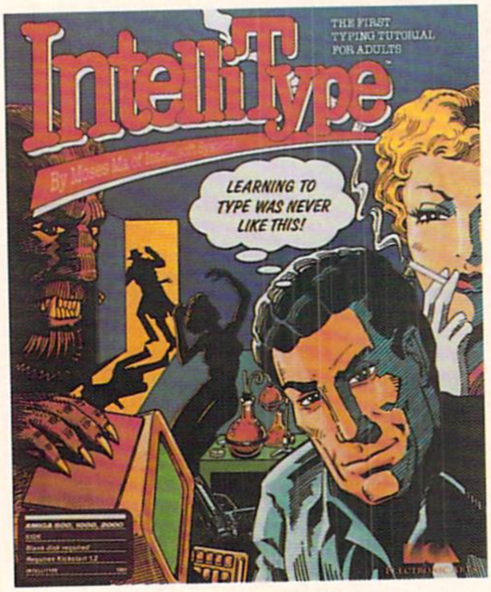
Tidy Computing

We all know the evils of CCAS—Crowded Computer Area Syndrome. Alongside your Amiga console, you've got a mouse, a 20Mb hard drive, an external 3.5 inch disk drive, scattered manuals, and a snarl of cables. Not only do you have to put up with an overcrowded desk, but there is also the danger of blocking the ventilation on your machine and hard drive.

Just when you thought you were doomed to this less-than-efficient environment, along comes the **Amiga 500 Command Center** from KETEK, Inc. to save the day ... and a whole lot of space and frustration.

For easy access, the Command Center allows you to sit either two 3.5 inch drives or one 3.5 inch and a 20 or

(continued)


IntelliType
THE FIRST TYPING TUTORIAL FOR ADULTS
LEARNING TO TYPE WAS NEVER LIKE THIS!
AMIGA 500, 1000, 2000

EA's IntelliType

30Mb hard drive directly above and behind your keyboard. Not only are the drives now out of your way, but they are also better protected—the Command Center includes a built-in cooling fan. Other useful features of the Command Center include surge supression, EMI noise filter, and a five outlet power strip which puts switches for each of your components at your fingertips. And if a well-dressed computer desk is your thing, the Command Center is color coordinated to match your Amiga 500.

Accessories are available to complement the Command Center and complete your workstation. Add a 20 or 30Mb hard disk, a one or two Mb internal RAM upgrade, a heavy-duty replacement power supply, joystick extension cables, a dust cover for your entire unit, or a tilt/swivel monitor stand. If you're frustrated by the look and inefficiency of your work area, let the Amiga 500 Command Center clean up your act.

Amiga 500 Command Center
$179 and up
(depending on accessories)

$99 for cabinet without electrical
components

KETEK, Inc.
P. O. Box 203
Oakdale, IA 52319
(800) 626-4582

Big Bucks!

Now that I've got your attention, I can tell you that big bucks is really what **Lottery Magic** from Slipped Disk, Inc. is all about. The program calculates and picks lottery numbers based on past results. You configure the program to fit your state's individual lottery, and Lottery Magic flashes the dream-deflating odds. I set up a file for the "pick six of thirty-six" lottery I play twice a week, but I still can't understand why I haven't hit the big one—my odds are a mere 1,947,979 to one!

Lottery Magic supports three and four-digit daily lotteries, and the multi-million dollar weeklies. You simply enter the numbers as they come up, and the program takes over. The number generator evaluates the results and comes up with a wallet-wrecking number of different ways to play.

You can play the "hot" numbers—the most common—or the "cold" numbers—the least common. If you want to get even more scientific, the generator notes the numbers that deviate the most and least from the norm. For good ol' fashioned players who rely solely on luck, the generator spits a random number and a mysterious "magic" combination. Lottery Magic also offers "systems" that wheel the various number groups to produce a set of related combinations you might want to try. How can you miss?

To make things easy on your harried gambler's mind, Lottery Magic develops graphs of much of the information. A data disk of results for each state will also soon be available from Slipped Disk, Inc. Lotteries are always long shots, but I, for one, will try anything to up my chances.

Lottery Magic
$24.95

Slipped Disk, Inc.
31044 John R
Madison Heights, MI 48071

Just Your Type

If you've ever taken a typing course, you know just how boring and repetitive the drills can be. It always seemed fruitless to type endless rows of "fjfjfjfjfjfj." Electronic Arts' new **IntelliType** typing tutor takes a shot at ending this drowsiness once and for all. The program combines an adventure story with artificial intelligence to teach lively typing in one month.

EA's philosophy is that technology is worthless if users don't finish the lessons. Enter the adventure, which the press release interestingly describes as "a mixture of *Soap* (the television show) and a James Bond film." As you learn to type, you hang with Ted and Laura through car chases, Bond-like stunts, and precarious meetings within foreign spies. A dash of wit and a pinch of sensuality put boredom into its own deep sleep.

The adventure is only part of the package—remember, your goal is to be an error-free blur at the keyboard. Unfortunately, practice is what makes perfect and drills have to figure in somewhere. You choose whether to emphasize speed, accuracy, or both, and IntelliType suggests drills based upon an analysis of your strengths and weaknesses.

The artificial intelligence of IntelliType (hence the title) allows the program to focus on drilling. 27 specific types of errors in nine categories are watched closely ... and specific is not the word. Errors related to dyslexia, doubling, high keystroke speed variance, transposition errors within or between hands, and even hitting the wrong shift key are all picked up and analyzed.

The analysis also deals in depth. Bar graphs reveal accuracy, speed, and error types and can be expanded to elaborate on each individual error type. An Accuracy Analysis Window presents percentages of near misstrokes, far misstrokes, and latency. Who says typing has to be boring?

IntelliType
$49.95

Electronic Arts
1820 Gateway Drive
San Mateo, CA 94404

Just Your Type, Too!

The Software Toolworks has also joined the fray for control at the keyboard with its own typing tutor, **Mavis Beacon Teaches Typing.** The concept is familiar—take the boredom and useless drill work out of learning to type.

In Mavis Beacon, we have a bit more of the classroom atmosphere than IntelliType, with a positive twist. Stepping away from statistics, this program follows the old school of learning—there is no better teacher than a teacher at your side. Mavis Beacon is a typing guru who explains your errors in conversational form. She even throws in a few jokes, historical quotes, and facts from the *Guiness World Book of Records* to stave off boredom.

Like IntelliType, Mavis Beacon also uses artificial intelligence to get the most out of lessons. 2-D and 3-D graphics and a pair of "Guide Hands" give you a visual display of perfect technique. Color graphs track your progress, and typing games—such as the arcade-like "Road Racer" and the musical "Tempo Typing"— help crack down on boredom.

Like any good teacher, Mavis Beacon shows how your typing experience fits into the real world. A "Resume Writer" program is included, and you can even get a jump on the actual tests used by employment agencies. Overall, Mavis Beacon offers back-to-basics instruction with a flavor that still conquers boredom.

Mavis Beacon Teaches Typing
$44.95

The Software Toolworks
Distributed by Electronic Arts
1820 Gateway Drive
San Mateo, CA 94404

Come Out Of Your Shell As A Programmer

The wall that stands between the user and the power of his computer is the interface. A programmer should have easy, intuitive access to the operating system, thus providing a cozy programming environment. Unfortunately, the wall between the programmer and the computer too often seems to be one thousand feet tall and made of iron. The ingenious answer to simplifying the interface has been the programmer's "shell." The latest entry in this market is **Tshell** from Metran Technology.

Tshell provides an advanced, but nicely simplified programming environment useful in software development or any other work that begs for full control of AmigaDOS. As a clone of the famed UNIX shells, Tshell acts as an "interpreted programming language," according to reps at Metran Technology.

APL.68000
$99

A HIGHLY OPTIMIZED ASSEMBLER BASED APL INTERPRETER FOR FAST AND POWERFUL PROGRAMS. FEATURES A COMPLETE INTERFACE TO THE AMIGA ENVIRONMENT WITH PULL-DOWN MENUS, REQUESTER AND ALERT BOXES, SPEECH, SOUND AND GRAPHICS FACILITIES.

Order direct for $99 + $7 shipping, $10 Canada.
VISA/MC/AMEX + 4% NJ res + 6% sales tax.

SPENCER
ORGANIZATION, INC.
P.O. Box 248 Westwood, N.J. 07675
(201) 666-6011

Tshell incorporates many Amiga-adapted features of the Unix environment. In addition to a Unix-like interface, Tshell also accepts Unix and AmigaDOS file names and Unix-style filename expansion. The syntax is similar to C, and command line editing and 39 built-in commands are offered. Other features include eight forms of I/O redirection, integer and string variables, 33 arithmetic and logical operators, four looping statements, and two conditional statements. Shorthand for current and parent directories is supported, and directory listings can be sorted in listed column form.

TShell also allows you to define your own function keys and initialize files for individual startup modes. Documentation is on-line for easy programmer's access and updates are promised based on user suggestions—I like that style. It's up to you to tear down the programming wall between you and your Amiga ... and Tshell might be a good place to start.

Tshell
$50

Metran Technology
P.O. Box 890
West Oneonta, NY 13861

•AC•

AmigaTrix III

A series devoted to making life with the Amiga cheaper, easier, and more fun.

by Warren Block

Newer readers of *Amazing Computing* may be confused by the title of this article because they have not seen the two earlier installments of AmigaTrix (in Episode One, we learned that Mavis had accidentally spilled nuclear waste into the scrambled eggs; in Episode Two, Arnold used his newly-grown tentacle to abscond with the Foogelvie Diamond).

Anyway, all the articles in this series are devoted to making life with the Amiga cheaper, easier, and more fun. Since the hints presented here are completely without pattern, it will be left up to the reader to assign some order (or indeed, some meaning) to them. Note the opinions expressed here are my own, and not those of *Amazing Computing*.

The Evils Of Copy Protection

Some software suppliers have had pity on Amiga owners. Rather than copy-protect their software with a DOS-based scheme, suppliers used an innocent-looking little hardware device called a "dongle." How thoughtful of them. These things usually plug into the second joystick port, and are just small enough to lose easily.

The solution to this problem is to travel down to the nearest sewing goods or department store and pick up a yard or so of half-inch wide stick-on Velcro (get both hook and loop types, or you'll get very frustrated). Cut a three-inch piece of the loop type and stick it onto the top of your computer or monitor in a readily accessible area. By placing strips of the hook-type material onto the dongles, you can stick them in place on your computer where they won't get lost. Marking both the dongles and their associated disks with dots of paint or colored markers will make them immediately identifiable.

Just to keep the Velcro people happy, I should also mention all of the other fantastic possibilities this stuff has. Stick it on the back of a calculator and hang it from the side of your monitor (despite software calculators and multi-tasking and all that hoo-ha, a real calculator is still handier); use it to route all the cables coming out of the back of the computer; stick long-haired hampsters to the ceiling with it. No, not really—just kidding! Heh, heh.

Of course, the best way to avoid dongle troubles is to simply avoid buying software that uses this copy-protection method in the first place.

Fast Copies

It can take an amazingly long time for the Amiga to print a page-sized picture. Multiple copies take just as long as the first one. But a public domain program called CMD (written by Carolyn Scheppner, one of Commodore's resident geniuses), trims the time down to a bare minimum.

CMD redirects into a file what normally would be printer output. Simple enough. You "print" the picture; all of the printer commands and data go into a file. Afterwards, just Copy the file to the PAR: device. Surprise! The printer rips off a print of the picture in record time, since it need not wait for the Amiga to generate and print every line, combined with the attendant slowness of the of printer device. (My Canon PJ-1080A took seven minutes to print a picture directly from Aegis Images.) Once the data had been captured in a file, however, the task was completed in only six minutes. Times will vary depending on the printer and the graphic image, of course.

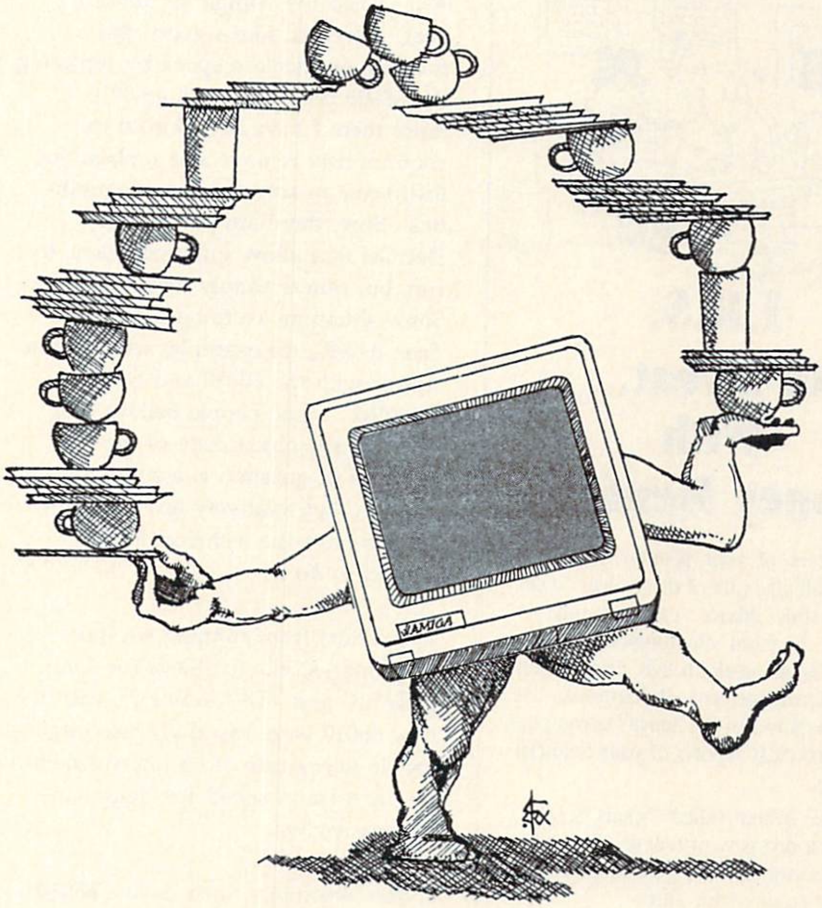
Note: when capturing a picture with CMD, be sure to use the "-s" flag. As the first part of a printout, a reset code is sent to the printer. When this code is received, the printer pauses long enough to lose a few of the following data bytes, resulting in a garbled picture. With the flag, CMD simply discards ("skips") the reset code, so it isn't included in the file. For example:

```
Run CMD -s parallel ram:picture.prt
```

redirects the output into the file called "picture.prt" in the RAM: device.

This trick also works with text to be printed, although it may not improve printing time, unless the program takes a long time to format output. If you wish to edit the output or import it into another program, it works very well.

Capturing text output this way usually produces a large and unwieldy file. Among its many shortcomings, ED (the AmigaDOS screen editor) has problems working with such files. Aedit (from DRM Programs, 1329 Arthur Avenue, Las Vegas, NV 89101, 702-457-9489) works well with huge files, and makes ED look like what it really is: slow and wimpy. See my review of Aedit in Volume 2, Number 11 of *Amazing Computing* for more details.



Rodent Revulsion

In a lot of situations, the mouse just makes things difficult. (Like in the database program where you have to click on a requester after entering every record.) Moving your hands from the keyboard to the mouse slows things down. However, in many such programs, the mouse pointer doesn't need to move—just click the Select button. The Amiga's been out long enough for early buyers to forget you can simulate the mouse with the keyboard. Press the left-side Alt and Amiga keys at the same time, instead of reaching for the repetitious rodent.

Some programs shut off the mouse pointer to get it out of the way, usually after some mouse move has been made. You could press Select or the left-Alt and Amiga keys, but in a word processor, that might move your cursor to the wrong spot. Pressing the Menu button or the right-Alt and Amiga keys shuts off the pointer without changing what you're doing in such situations.

Creature Comforts

When you sit down at the keyboard, you're likely to be there for at least an hour. Many people complain that looking at any monitor screen (even those of expensive, super-sharp RGB monitors) for that length of time causes their eyes to hurt. Coincidentally, these same people tend to be the ones who wipe the dust off their screens once a year. Dust is attracted to video screens like moths are attracted to porch lights, and it causes the displayed image to become blurry, which in turn causes severe eye fatigue.

Don't get the idea I'm advocating attacking your monitor with a spray cleaner; just use a dry, soft, clean cloth. If you really must have a liquid, use plain water. Of course, your monitor should be off at such times. The "anti-static" sprays and cleaning supplies sold for this purpose are usually unnecessary and expensive.

Not only can dust affect the sharpness of your screen; airborne dust and smoke particles grind between disk media and the heads every time you use them. With furnaces running in the winter, heavy deposits can accumulate in a short time. The easiest way to protect your investment is to prevent dust from accumulating around your computer in the first place. When you clean the screen as suggested above, wipe the dust off the top and sides of the monitor, the exposed top of the computer and any external disk drives, and the tabletop surrounding it. Don't leave disks laying out in the dusty air; keep them in storage containers. A towel can be used to cover the mouse and internal and external disk drives at the same time. Another can be used for a printer. Or you can pay $20-$30 for vinyl anti-static dust covers that serve the same purpose.

When you are ready to use the computer, remove the covers and set them somewhere out of the way—NOT on top of your monitor. The covers can trap unwanted heat in the monitor.

Other factors affect your comfort, too. Location and height of the keyboard can be very important, although these factors are much easier to adjust on the A1000 and A2000 than the A500.

(continued)

I.R.S.
No Sweat...
With
Money Mentor™!

"Keep track of your pennies, and your dollars will take care of themselves." Old, but sensible advice, even in today's complex financial environment. Money Mentor™ is a breakthrough in personal financial management. It harnesses the awesome power of the Amiga™ to compute and graph clear reports of your financial situation.

A unique system called "Smart Scrolls" handles a diversity of tedious data entry functions and can save 70% of the typing typically required for entry.

Money Mentor™ features:

- 200 budget categories.
- 30 integrated accounts: checking, cash, saving and credit cards.
- Elaborate search routine allows editing of transactions according to your specific guidelines.
- Automatic check printing.
- Automatic Account Balancing.
- Colorful graphic reports illustrating actual versus budgeted amounts.
- Over 50 reports from which to choose.

This year... get organized with Money Mentor™

SEDONA SOFTWARE
11844 Rancho Benardo Rd., Ste. #20
San Diego, CA 92128
To order,
call (619) 451-0151
MasterCard VISA

Is It Really Ten Better?

After I had my Amiga for about a year, it seemed like a good idea to moderately increase speed by replacing the 68000 processor with a 68010. Since then, I have been forced to continuously remove and replace the 68010 due to software incompatibilities. Sure, there are patches like DeciGel that allow some programs to run, but others simply refuse to work. Some situations were nearly as bad. True BASIC, for example, actually ran slower with the 68010 and DeciGel installed. Some people believe that patching the object code of the program in question is a simple solution, but relatively few Amiga owners have the technical ability needed to do so.

The results from running several benchmarks, like the Sieve (in AmigaBASIC and TDI Modula-2), with the new 68010 were less than encouraging, too: in every case of an improvement, the increase in speed was less-than-impressive 3% .

If you absolutely must have a 68010, be aware that you will have to deal with these problems. Removal of the 68000 is not a job for the inexperienced, either—I've seen too many mangled chip pins and people with neat rows of holes punched in their thumbs to recommend the procedure to novices. Once you've fought your way past all the screws and RF shielding and actually removed the 68000, be sure to put it in a piece of conductive foam in a safe place—you'll most likely need it again.

For most people, a 3% increase in speed just isn't worth the hassle of fighting with the 68010.

Labeling Things

Dot-matrix printers are popular, and with good reason. They are versatile enough to print both graphics and text and can use various preprinted forms, even print mailing and disk labels.

Unfortunately, printed output can suffer when printing graphics and labels, usually from slippage in the tractor feed gearing. The result can be a graphic dump with white lines running through it, or jagged, misaligned characters.

Then there are those printers which have tractors that can't be positioned close enough to engage the schnybbles (that's what those tear-off pinfeed rows are called) on both sides of a label stack. When labels are pulled by only one tractor, the feed is often crooked. Sometimes they get off-track enough for a label to peel off the backing sheet and stick to the platen. Unless you like to disassemble consumer electronics items with lots of small screws, springs, and unnameable whatzits, this is not something you want to happen to your printer. (If the unthinkable should happen, dissassemble the printer and remove the platen. Then you can use isopropyl alcohol to clean the sticky gunk off.)

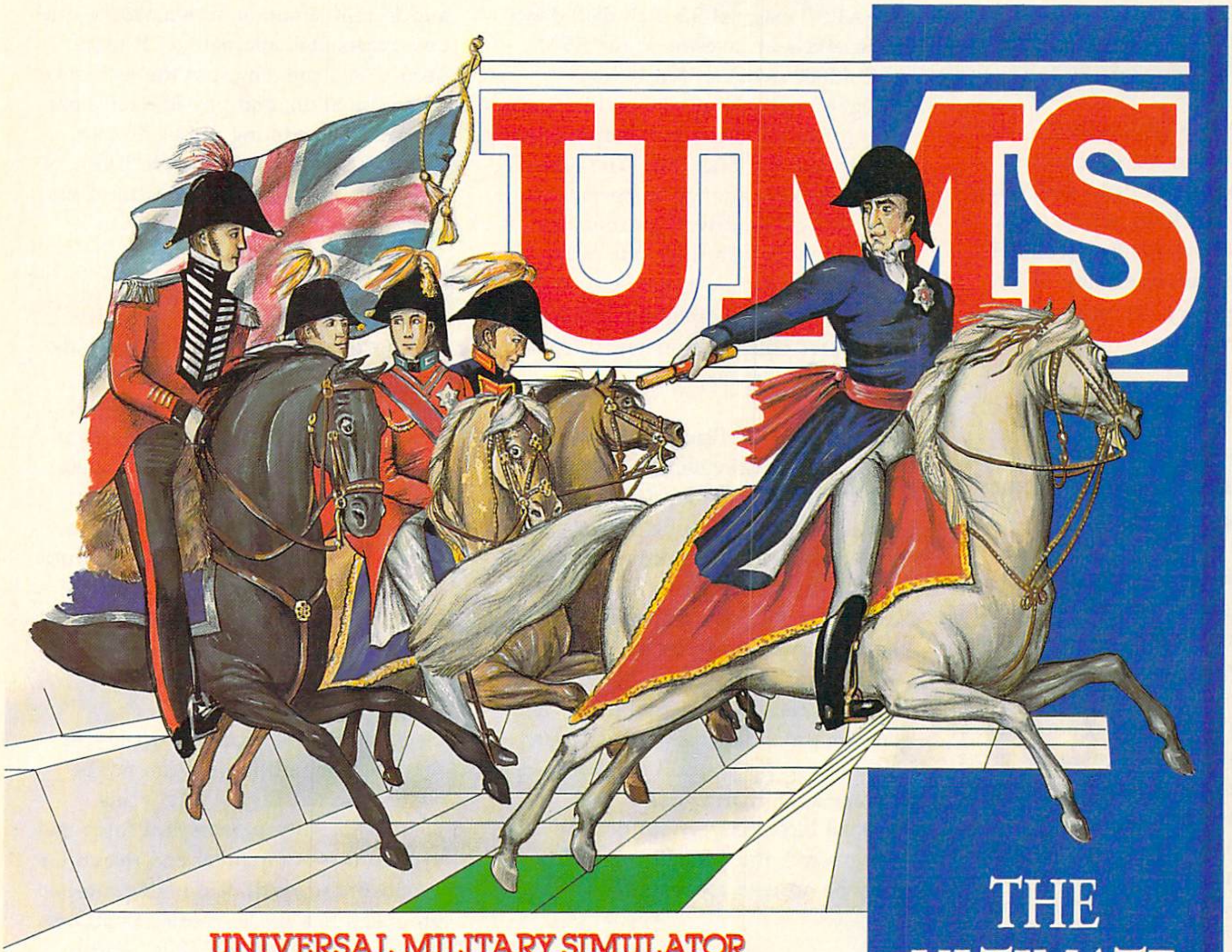
Surprisingly, both problems can often be corrected by the same adjustment. Just load the labels (or paper for a graphics dump) and turn on the friction feed on the printer. On most modern dot-matrix printers, the tractors will still pull when the friction mode is on. The extra force will help keep slack out of the gearing, making for unmarred graphics dumps. It will also keep labels feeding straight, because they won't be pulled (or pushed, depending on your printer) on only one side.

Sort Of

Everyone occasionally needs a printed disk directory. To make things easy to find, a sorted listing is best. With only the tools provided by an ordinary Workbench disk, this is an easy task.

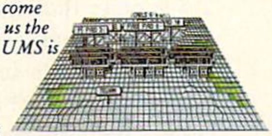
First, CD to whatever directory you wish to print out. Then use output redirection to send the output of the

(continued on page 40)



UNIVERSAL MILITARY SIMULATOR

You may never buy another war game.
"One's first impression of UMS is that of awe. The computer-generated topographical battlefields, the highly playable nature of the game, and the very flexible nature of the construction set come together to give us the impression that UMS is definitely a breakthrough product."

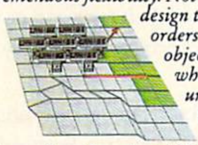


That's what the wargame experts at Computer Gaming World say about UMS. They also like UMS' high-tech graphics...

"...the three-dimensional topographical views of the battlefield offer a unique perspective in computer gaming. The ability to look at the entire map from eight different compass points is, to our knowledge, unique."

...the power and flexibility...

"...the program's constructability allows for tremendous flexibility. Not only can players design their own maps, orders of battle and objectives, but players who feel that certain units are too strong, too fast, or too experienced can adjust those with the powerful editor...the program truly does allow almost any historical, fantasy or science fiction battle to be created."



Screenshots from Atari ST.

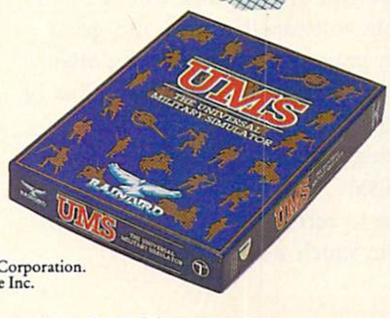
"...gamers will appreciate the program inclusion of elements such as terrain, elevation, weaponry, morale, status, and unit efficiency in its calculations."

...and to sum it all up...

"...we want to say that this is a cutting edge product that all computer wargamers will want to play."

Universal Military Simulator. It's the new standard in wargames.

Available now on Atari ST and IBM. Coming soon on Amiga and Mac.



Atari ST© Atari Corporation.
Mac© Apple Computers Inc.
IBM© International Business Machine Corporation.
Amiga© Commodore Business Machine Inc.

Telecom Soft, P.O. Box 2227, Menlo Park, California 94026.

THE
ULTIMATE
MILITARY
SIMULATOR

Intergalactic
Development
Incorporated

RAINBIRD

Rainbird and Rainbird Logo are registered trademarks of British Telecommunications PLC

(continued from page 38)

List command to a file in RAM:, like so:

```
List > RAM:temp
```

Load the file into ED (or Aedit—but I did say you could do all this with an ordinary Workbench disk) and delete the first and last lines, which show information about the listing rather than actual directory entries. Then sort the temporary file and print it at the same time with:

```
Sort RAM:temp to PRT:
```

You can use SER: or PAR: instead of PRT:, or a disk filename to save the sorted listing for later use.

Note that the Dir command won't work for this purpose, for reasons not immediately obvious to most people. Try it out to see why.

Another use of the Sort command is to create a miniature database. By using one-line entries, and putting the sort field at the front (or using Sort's "colstart" flag) is easy. For example, a group of names entered in the following format would be sorted by name:

Jones, Bill, 888 Eigth, Sippleville, NC, 88088 (808) 808-1234

while a file in the format

(808) 808-1234, Jones, Bill, 888 Eigth, Sippleville, NC 88088

would be sorted by phone number. Sort sorts by ASCII value, so some entries may not be in the obviously correct place, but they will usually be close enough to make it simple to correct them with a text editor. The biggest problem with using Sort this way is that everything must be on one line. Since some text editors have line length limits of 512 characters (Aedit does, and I believe TxEd does also). It's not such a bad problem after all.

Limiting Factors

An A1010 external 3.5-inch disk drive is an excellent investment for A500 and A1000 owners. But it was designed with the A1000 and its front-facing internal drive in mind, making the drive a bit inflexible. The cable is not quite long enough to permit easy selection of the drive's location; for A500 owners with their side-facing internal drives, the situation is even worse. A longer cable would be nice. But so far none have emerged from the factory this way.

Fortunately, Jim Black of BCD (P.O. Box 1224, Bartlesville, OK 74005, 918-336-1784) has two-foot drive extension cables for only $13 (plus $2 shipping), which eliminate the need to stand on your head to load disks into your external drive. A number of other hard-to-find accessories are also available from BCD, including power supplies for extra external drives and custom monitor cables. If you want to wire things up yourself (What could be more fun than wiring a cable that, if done incorrectly, could destroy your computer?), the oddball connectors may be ordered alone, too.

Bandwidth Considerations

On-line networks like PeopleLink and CompuServe have made access to information and new public domain software much easier for Amiga users, but before you download that program, consider the cost. Some Amiga programs (especially ANIM format animations) can take two or three hours to download. At roughly five dollars per hour, you've spent fifteen bucks for one program.

Or you can order a disk full of public domain software by mail, and get a dozen programs or so for six dollars (assuming, of course, that you already subscribe to *Amazing Computing*—if you don't, you'll pay an extra dollar per disk). You will have to wait a bit longer to receive the disk, but it will contain much more software.

You may need a program urgently, and in that situation, downloading can be a reasonable alternative. If your need is less pressing, use the networks for information, and pay less for more software by ordering public domain disks. You'll find a complete listing of more than 130 (!) public domain disks in this issue.

Evolution At Work

My Startup-Sequence file is constantly modified. Almost daily, new commands are added, others deleted, still others modified. But a few remain in place constantly. I may change their parameters a bit now and then, but the programs themselves are always there. These lean and hardy survivors have come up against contenders many times, but they have always prevailed (so far). These "fittest" programs have a lot in common.

One simple, but necessary quality, is that the program in question not be copy-protected in any way. This includes those programs that force me to hunt down a manual and search for a keyword, as well as dongle-protected programs like those mentioned above. The delay just isn't worth it. Fortunately, really good programs are never copy protected.

Traits other than lack of copy protection play just as important a part in a utility program's survival. The program shouldn't adversely affect system performance. If things slow down, or it gets in the way, it's history. Programs that want input during a long booting process should at least be able to get it from a redirected command file (using the "<" symbol); of course, a command-line option is handier. Finally, the program should disappear completely, unless I summon it somehow.

The programs that have managed to survive in my Startup-Sequence so far include:

(continued on page 42)

TAKE A DRIVE INTO TOMORROW

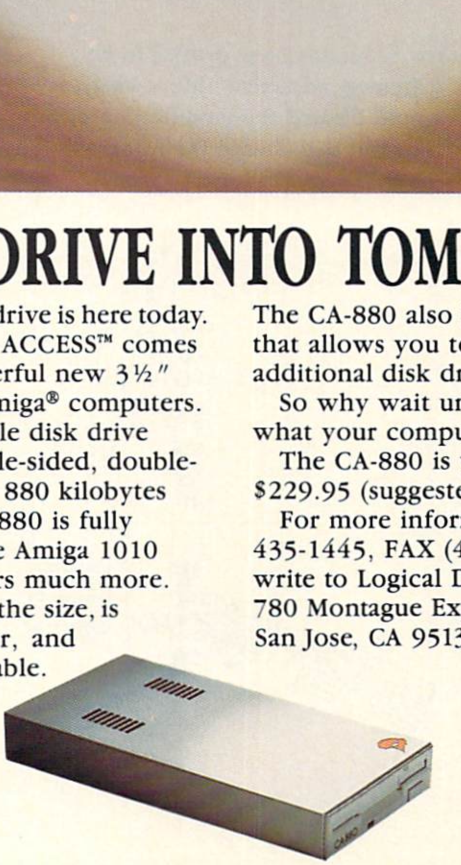
Tomorrow's disk drive is here today. From CALIFORNIA ACCESS™ comes the CA-880, a powerful new 3½″ disk drive for all Amiga® computers.

This highly reliable disk drive formats a 3½″ double-sided, double-density diskette for 880 kilobytes of storage. The CA-880 is fully compatible with the Amiga 1010 disk drive, but offers much more. The CA-880 is half the size, is considerably quieter, and has an extra long cable. The CA-880 also has a connector that allows you to attach an additional disk drive.

So why wait until tomorrow for what your computer needs today.

The CA-880 is yours for only $229.95 (suggested retail price).

For more information call (408) 435-1445, FAX (408) 435-7355, or write to Logical Design Works, Inc., 780 Montague Expwy., #403, San Jose, CA 95131.



CALIFORNIA ACCESS ™

Your Access to the Future

Amiga is a registered trademark of Commodore-Amiga, Inc. CALIFORNIA ACCESS and CA-880 are trademarks of Logical Design Works, Inc.

ADD TO THE POWER OF YOUR PROGRAMS WHILE YOU SAVE TIME AND MONEY!

CBTREE does it all! Your best value in a B+tree source!

Save programming time and effort.
You can develop exciting file access programs quickly and easily because CBTREE provides a simple but powerful program interface to all B+tree operations. Every aspect of CBTREE is covered thoroughly in the 70 page Users Manual with complete examples. Sample programs are provided on disk.

Gain flexibility in designing your applications.
CBTREE lets you use multiple keys, variable key lengths, concatenated keys, and any data record size and record length. You can customize the B+tree parameters using utilities provided.

Your programs will be using the most efficient searching techniques.
CBTREE provides the fastest keyed file access performance, with multiple indexes in a single file and crash recovery utilities. CBTREE is a full function implementation of the industry standard B+tree access method and is proven in applications since 1984.

Access any record or group of records by:
- Get first
- Get previous
- Get less than
- Get greater than
- Get sequential block
- Get all partial matches
- Insert key and record
- Delete key and record
- Change record location
- Get last
- Get next
- Get less than or equal
- Get greater than or equal
- Get partial key match
- Get all keys and locations
- Insert key
- Delete key

Increase your implementation productivity.
CBTREE is over 6,000 lines of tightly written, commented C source code. The driver module is only 20K and links into your programs.

Port your applications to other machine environments.
The C source code that you receive can be compiled on all popular C compilers for the IBM PC and also under Unix, Xenix, and AmigaDos! No royalties on your applications that use CBTREE. CBTREE supports multi-user and network applications.

CBTREE IS TROUBLE-FREE, BUT IF YOU NEED HELP WE PROVIDE FREE PHONE SUPPORT.
ONE CALL GETS YOU THE ANSWER TO ANY QUESTION!
CBTREE compares favorably with other software selling at 2,3 and 4 times our price.
Sold on unconditional money-back guarantee.
YOU PAY ONLY $99.00 - A MONEY-SAVING PRICE! NOW! Variable length records.
TO ORDER OR FOR ADDITIONAL INFORMATION
CALL (800) 346-8038 or (703) 847-1743
OR WRITE

PEACOCK SYSTEMS, INC.

Peacock Systems, Inc., 2108-C Gallows Road, Vienna, VA 22180

(continued from page 40)

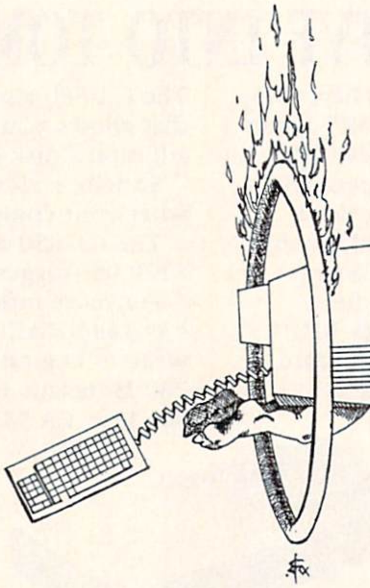
•StarTime—Sets the Amiga system clock from the MicroBotics StarBoard2 MultiFunction Module. Included with the MFM package.

•FastFonts—Speeds up text output up to six times, improving the performance of any text editor. From MicroSmiths, Inc., makers of TxEd.

•ConMan—A shareware program from William Hawes that makes the CLI truly usable. If you don't already have ConMan, get it. On a PD disk near you, or from Fred Fish Disk #100. Be sure to send Mr. Hawes a well-earned donation of at least $20.

•Facc II—A disk-caching program that speeds AmigaDOS file access, and reduces the gronking of disk drives. From ASDG, Inc., long known for their Amiga expansion hardware.

If problems with one of these programs should appear in the future, I may have trouble abandoning it—all the programs have proven to be extremely addictive. More likely, I would discard the modification or new program that caused the trouble in the first place.



Before you buy an add-on utility, see if you can try it out to verify that it is as well-behaved as these programs. Call the manufacturer if your dealer is unable to answer your questions. By sticking with programs of this quality, your time can be used for productive work, rather than fighting a multitude of irritating incompatibilities.

In Closing

Try to keep in mind that these ideas are not meant to be the final word. Instead, they should be starting points for your own adventures. Good luck!

Those interested in corresponding with this Block weirdo about this or any other article may write to him care of this magazine. Although replies are not guaranteed to all—SASEs help. Replies that do materialize are likely to be pretty strange.

•AC•

EXTEND

An AmigaBASIC Extension

by Bryan Catley

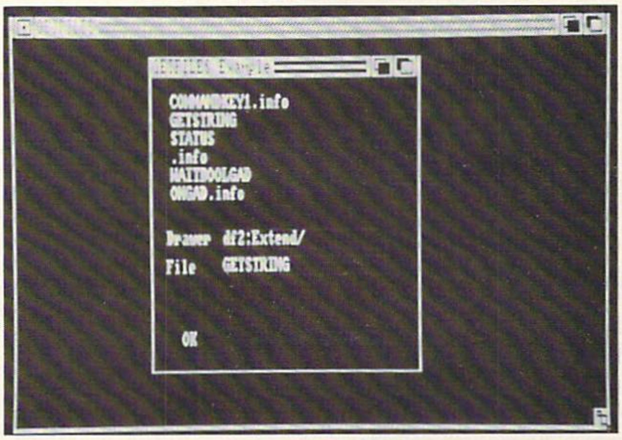
Despite its ability to access operating system routines via the CALL statement, AmigaBASIC does not provide easy access to many of the Intuition routines. For example, it is very difficult, if not impossible, to access the built-in facilities for using requesters, gadgets, sub-menus, changing gadget colors, changing menu colors, changing window titles, etc. But now you can use all of these facilities, and more, with a EXTEND, a new product for AmigaBASIC users.

EXTEND is implemented with the standard LIBRARY command (which means it also has an associated ".bmap" file), is extremely easy to install, and provides users with 33 new commands with very little additional overhead! It may also be used with interpreted programs and programs to be compiled with Absoft's Basic compiler.

The Additional Commands

What are these additional commands? Well, just to let you know exactly what's there, let's list all 33 of them:

BUSY	toggles pointer between standard and "WAIT"
CHANGE	changes window titles
CHECK	polls a series of gadgets
CHECKSUB	checks for sub-menu selection
CLEARBUFFER	clears contents of a string gadget
CLOSELIB	de-allocates gadget memory
COMMANDKEY	sets a command key sequence for a given menu item
DOS	executes a valid AmigaDOS command
FREEMEM	de-allocates all memory allocated via MEMORY
GADBCOLOR	sets attributes for gadget borders
GADGET	places a gadget in the current window
GADTCOLOR	sets attributes for gadget text
GETINPUT	activates a string gadget
GETFILES	displays a scrollable list of file names
GETSTRING	returns the contents of a string gadget
MAT	changes hilighting attributes for a menu item
MEMORY	allocates a block of system memory
OFFGAD	disables a given gadget
ONGAD	enables a given gadget
PROTECT	protects a string variable from system manipulation
PUTSTRING	places a string variable into a string gadget
REFRESH	refreshes current output window
REMOVEGAD	removes a gadget from the current window
REQCOLOR	sets color of text in requesters
REQUEST	prompts user with a requester
SETWINDOW	provides EXTEND with window information
SREQUEST	prompts user with a custom string requester
SREQCOLOR	sets color of text in a custom string requester
STATUS	determines the status of a string gadget
SUBCOLOR	sets color attributes for all menu sub-menus
SUBMENU	assigns a sub-menu to a given menu item
WAITBOOLGAD	waits until a given gadget is clicked
WAITGAD	waits until a given gadget is not active



```
COMMANDKEY1.info
GETSTRING
STATUS
.info
WAITBOOLGAD
ONGAD.info

Drawer df2:Extend/
File   GETSTRING

  OK
```

The most interesting command is probably GETFILES. This single command will open a window with a given title, allow you to select any path, then display the list of files, which may be scrolled up and down within that path. It is also possible to return to the parent of the current path if desired. When a file name is finally selected, it is returned to the using program, along with its full path. The file may then be processed as desired—a very powerful command!

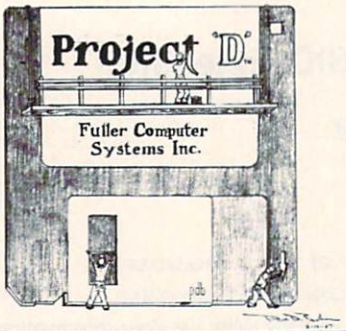
Using These Commands

If you have ever used any operating system call on the Amiga, such as SetSoftStyle, Move, TextLen, etc, you already know how to use these new commands. It's that simple!

You must provide a LIBRARY statement for the EXTEND library, and a DEFINE FUNCTION statement for each command which will be used to return some value to the using program. From then on, just code the desired commands and parameters.

(continued)

INTRODUCING.....



An
Evolution
in Disk
Utilities
for Amiga™
Personal
Computers!

FEATURING
- An easy to use, friendly and intuitive user interface.
- A powerful and fast disk backup tool that lets you make backups of your copy-protected Amiga software.
- A disk editing tool that lets you edit raw MFM tracks, AmigaDOS sectors and AmigaDOS files (automatically calculating new checksums).
- A disk cataloging tool that lets you maintain lists of your personal, public domain and commercial software.
- A unique backup tool for duplicating other disk formats including MS-DOS/PC-DOS and Atari ST.
- An easy to read, informative user manual is included.
- This product is not copy-protected in any way.

NOW SHIPPING!
$49.95 EA.
Includes shipping and handling!
Arizona residents add 6.5% sales tax.

TO ORDER
Send check or money order to:
Fuller Computer Systems, Inc.
P.O. Box 9222
Mesa, Arizona 85204-0430
Or CALL (602) 835-5018

Amiga is a trademark of Commodore-Amiga, Inc.
Dealer Inquiries Invited

However, there are two things which you should be aware of. First, one mandatory command, SETWINDOW, must be used at the beginning of the program and each time the program switches windows. Secondly, some functions require a series of commands, in a specific sequence, to be accomplished successfully.

A short and simple example might be:

```
  DEFLNG a-Z
LIBRARY "extend.library"
SETWINDOW WINDOW(7)
DOS SADD("list df1:"+CHR$(0))
CLOSELIB
LIBRARY CLOSE
END
```

Note the DOS command must usually be used from a program started via a CLI, and not via Workbench. This is because most AmigaDOS commands direct their output to the current CLI window, and there is no such window when a program is started via Workbench. When attempted, the result is almost always a system crash!

Distribution and Installation

EXTEND is distributed on a single, non copy-protected, bootable disk. There are only two files of real interest on the disk: "extend.library" and "extend.bmap." For the most part, the rest are standard Workbench files.

In addition to the disk, the package contains some documentation (more on this later) documentation updates (if any are required), a registration card, a q & a sheet regarding potential problems the user may encounter, a license agreement (for possible distribution of developed programs, I assume), and promotional material for other products.

Installation is very simple but must be done via a CLI. If you think you would like to use this product but have not yet used a CLI, you should familiarize yourself with it first.

The documentation suggests you boot your Amiga using the EXTEND disk, then copy the extend files to the disk of your choice. Since this disk will almost always be your normal working Workbench disk, I suggest you do things the other way around: boot your machine as usual, then copy the files from the EXTEND disk to your working Workbench disk. (Remember, only work with disk copies, not the originals.) Assuming you're using a two drive system with the EXTEND disk in DF1:, you can use a command such as:

```
COPY DF1:libs/extend#? TO DF0:libs/
```

You are now ready to start using EXTEND! However, you will probably want to run most of the examples before you start using the new commands in your own programs. The easiest way to accomplish this is to "drag" the Examples drawer from the EXTEND disk to your AmigaBASIC disk where you may double-click the examples to run them.

All the examples run, and many contain multiple commands showing how they should be used in conjunction with each other. If you have a printer, it would be worth your while to print all 32 of them. (They average 30 to 40 lines each.)

Problem Areas

Great as EXTEND is, it is not without a few problems. But if you have any programming experience at all, they should not be of serious concern to you.

One of the most obvious problems is that EXTEND is somewhat difficult to use. For example, all string parameters must be passed as addresses, making them awkward at best. (Wouldn't it be nice to just say DOS "list df1:"?) You should also note that commands are very sensitive regarding syntax. A missing ")", for example, will result in a system crash rather than a simple error message! (These two problems are shared with all calls to all library routines.)

Command sequence is also very important when several commands are required to accomplish something. For example, to display a gadget on the screen, you must first describe it in an array, the address of which is passed to the GADGET command via a basic VARPTR statement. Well, no basic statements are allowed between the VARPTR and GADTGET statements; in fact, you must use two separate statements. (It seems the VARPTR may not even be included as a parameter within the GADGET statement.)

While EXTEND is almost fully compatible with Absoft's BASIC compiler, there are differences in the way some things work. These differences are documented in the Q & A sheet and should not prove too serious; they may require two versions of the program, though. The one known incompatibility is that programs making use of sub-menus will not function correctly when compiled. Apparently, this is a problem with the compiler—since AmigaBASIC does not support sub-menus, the compiler is simply not set up to handle them. The problem should be fixed in an upcoming release. Nonetheless, this restriction is not documented.

The GETFILES command is a handy one, but with the version of EXTEND I am using, its output is directed to the Workbench screen, making it unusable with a custom screen. (Please note that a later version of EXTEND does allow custom screens with the GETFILES command.)

All numeric parameters used with extend commands must be defined as long integers. This means that either a DEFLNG a-z statement must be included at the start of the program (forcing all numeric variables to default to long integers), or they must have a "&" appended to them (this also includes the commands themselves). It is no different from any standard (non-EXTEND) operating system call, but it is still a little annoying for the AmigaBASIC programmer.

As you can see, these problems are more annoying than serious. Careful coding, and learning some of the more unusual requirements, will enable you to use the EXTEND package to its full advantage, with minimal frustrations.

The Documentation

EXTEND's documentation is probably its weakest area. It consists of 12 pages of reduced and copied text which is somewhat difficult to read. It is arranged with a small "Getting Started" section at the beginning, followed by a description of each command. The descriptions are (for the most part) reasonably complete, but they give very little indication of the interaction between the various commands. You have to search through the examples then hope you can understand what is going on. As already indicated, there are also a number of things omitted.

Project Disk Special | RAMTools | WBExtras
Open | CLI
Close | WBExecute
Load | ViewText
Unload | SeePicture
LoadPicture
UnloadPicture

WBEXTRAS
by Peter Dunlap

Isn't it time you got the most from your Amiga?

Now, "WBExtras" is here and is specifically designed to enhance operation of the Amiga by the "New User" as well as the "Seasoned Programmer".

For the Amiga User . . .
Use of New Workbench Menus, "RAM Tools" and "WBExtras" provide access to ANY Workbench Tool from the Workbench Menu and allow "Multiple Icon Selection" without the use of the "SHIFT Key". Also, "New Execution Modes" permit a "Single Loading" of Workbench Tools for Multiple Task Execution. This results in "Optomized Memory Allocation" and "Reduced Disk Thrashing". For FULL System Memory, WBExtras will "PolitelyRetire" and RELEASE ALLOCATED MEMORY WITHOUT RE-BOOT. As a BONUS, several New Workbench Tools are included (See Menu).

For the Amiga Programmer . . .
WBExtras includes SOURCE CODE in "C" and "AmigaBASIC" for Workbench Tools using a NEW Programming Technique which provides "Optomized Memory Utilization". "Inter-Program Communication", and "Disk Access Queing".

Lynn's Luna
SYSTEM INNOVATIONS
P.O. Box 1308
Cañon City, CO 81212
303 275-5858

$39.95
Plus 3.00 for Shipping
Colorado Res. Add Sales Tax

MasterCard VISA

Amiga & AmigaBasic Workbench TM of Commodore - Amiga, Inc.
Dealer Inquiries Invited

Ideally, a tutorial section explaining how things fit together should be included. Sections covering the use of Gadgets, Sub-menus, Requesters and so on, using the given examples as a basis, would prove very useful to many potential users. Don't be put off though. It just involves a little more effort on your part to understand things thoroughly!

Using EXTEND

When reviewing a product of this type, it is somewhat difficult and time-consuming to use every command in a custom programming environment, especially when programmed examples of every command are also provided. However, I did want to check out some of the commands in programs which use custom screens and windows. To do this, I modified some existing programs, concentrating on sub-menus, command keys, requesters, and gadgets—features which use most of the commands anyway.

Sub-Menus, *et al.*

The first effort was directed at modifying some custom menus to use sub-menus, box (rather than complimentary) highlighting, and command keys. After I studied the examples and documentation, the changes were relatively easy to make, and the results were excellent (although I would have liked to be able to choose where the sub-menus appear, rather than accepting EXTEND's arbitrary choice).

(continued)

Even up the Score !!!

INVESTOR'S ADVANTAGE

Start using your Amiga to give you the Advantage in making better investment decisions! Color Graphics of Individual Stocks and General Market Trends. High Low Close, Moving Averages, Volume, Momentum and Relative Strength. Select the best performers with Relative Strength Rankings. Know when to get into and out of the market. Update stocks and/or mutual funds manually or automatically. Easy to use Communications included.

Only $99.95

See your local dealer or call:

Software Advantage Consulting Corporation
37346 Charter Oaks Blvd
Mt. Clemens, MI 48043 (313) 463-4995

Amiga and The Investor's Advantage are trademarks of their companies.

Four commands are required to accomplish all this. SUBCOLOR defines the colors to be used when displaying the sub-menus; SUBMENU defines the actual sub-menus to be used; MAT defines main and sub-menu menu characteristics (type of highlighting, checked/non-checked, mutual exclude flags, etc); and COMMANDKEY defines the "right Amiga-key/key" combinations which may be used as alternates to the menus. I also made extensive use of the BUSY command, another really neat feature of EXTEND.

However, I did run into a couple of problems while doing this. First, as already mentioned, the documentation does not provide enough information. Second, EXTEND is very unforgiving of errors. Any simple syntax error will almost certainly provoke a visit from the Guru rather than a simple error message! (Note that this is true of any and all system calls, not just those for EXTEND.) Documentation omissions included: command sequencing (in this instance you must define all the basic menus first, provide a delay which allows AmigaBASIC to set them up, define the sub-menus, define the menu characteristics, and then define the command keys), the need to provide space in the menu descriptions for the command keys, how to initialize a menu item with a check-mark, and so on. However, the results were excellent and worth the effort!

Gadgets

As far as gadgets are concerned, I worked with the examples provided, setting up custom screens and windows, changing colors, etc. All worked well, but once again, there were a few frustrations of the type described above.

Gadgets are drawn as a single line box along with the associated text. You define the location and size of the gadget, with the location of its text in an array which is then passed to the GADGET command. With the present release of EXTEND, you may also choose between Boolean (Yes/No) and string gadgets. Proportional gadgets (sliders) are not yet supported. If you want "shadowed" gadgets, or gadgets with an additional outline, you must draw these extras yourself using standard AmigaBASIC commands.

Colors are set with separate commands for border and text. Each of these commands has three parameters: foreground color, background color, and drawing mode. However, the border color command does not seem to make any obvious use of the latter two parameters.

Setting up and using these gadgets is a lot easier than doing the entire thing in AmigaBASIC! Plus, a couple of additional features (like disabling and enabling gadgets) automatically become available to you. Very useful!

Program Development and Distribution

EXTEND, especially when used in conjunction with Absoft's AmigaBASIC Compiler, makes AmigaBASIC a really viable development language. (The biggest drawback is the excessive amount of memory the final product requires.) So, if you develop a program using EXTEND which you wish to distribute, it follows that you will need to distribute EXTEND along with your program. The owners of EXTEND will allow you to do this for a one-time fee of $100 per distributed program. It is a great deal, and if you have been waiting to do some development with a product like EXTEND, take advantage of this offer. (I feel that as EXTEND increases in popularity, the fee could increase too.)

Some Final Words

When it comes to products which extend the capabilities of an existing language, the bottom line is: "Should I buy it?" My answer is pretty straightforward: If you have been using AmigaBASIC for a while, if you are reasonably familiar with operating system calls, and if you feel you can make use of the additional facilities offered by EXTEND, buy it! I doubt you will regret it. Just be prepared to accept a few frustrations. If you do purchase it, remember to send in the registration card. By the way, there is a $5 charge for updates.

•AC•

EXTEND
by SunSmile Software
533 Fargo Ave. Buffalo, NY, 14213; (716) 885-5670
suggested retail price is $59.95.

AssemPro

by Stephen Kemp
Plink ID: SKEMP

The professional assembly language development system.

AssemPro Amiga from Abacus Software is billed "The professional assembly language development system." As the quote implies, AssemPro is more than a mere assembler. The package contains a source editor, symbolic debugger, and a tables (help) utility. The one unique feature of the package is that these development tools are integrated, allowing easy movement from one utility to another.

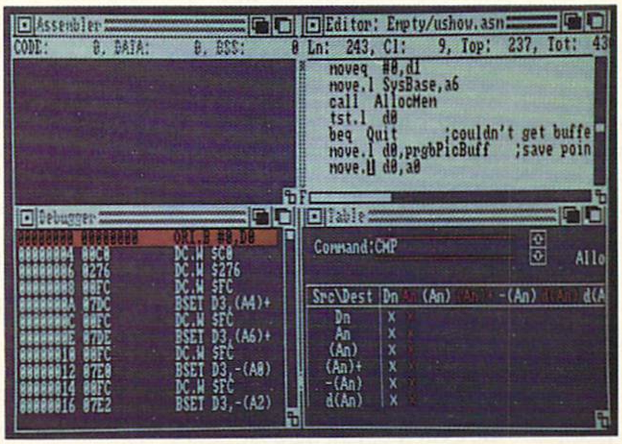
Although reference has been made to "utilities," it would be more accurate to describe the utilities as tasks. AssemPro is a single program that, when executed, initializes and opens a window for each task. This provides an excellent environment for program development. For the sake of clarity, I will continue to refer to these tasks as separate entities.

The System

The documentation accompanying the package is divided into sections describing the utilities' functions. It is strictly a reference tool for the menu options available in each utility. No summary of 68000 instruction syntax or Amiga operating system interface is provided in the documentation. Likewise, very few techniques and requirements of assembly language programming are documented. AssemPro assumes you already know these things or know where to get the information.

With these assumptions, section 1 of the documentation provides a short "guided tour." This introduction is meant to familiarize you with the development system. Although the example is a simple one, and does not provide any programming insight, it does demonstrate the interactive nature of the utilities.

Since most programs begin in an editor, this is where we will begin. The editor is fairly simple to use and provides most of the normal operations expected from an editor. Menu choices are provided for each of the editor's functions; many of them also have keyboard alternatives.



```
CODE:        0. DATA:        0. BSS:        0
```

```
Ln:  243, Cl:     9, Top:  237, Tot: 43
    moveq  #0,d1
    move.l SysBase,a6
    call  AllocMem
    tst.l  d0
    beq  Quit        ;couldn't get buffe
    move.l d0,prgbPicBuff    ;save poin
    move.l d0,a0
```

```
00000000 00000000      ORI.B #0,D0
00000004 00C0          DC.W $C0
00000006 0276          DC.W $276
00000008 00FC          DC.W $FC
0000000A 07DC          BSET D3,(A4)+
0000000C 00FC          DC.W $FC
0000000E 07DE          BSET D3,(A6)+
00000010 00FC          DC.W $FC
00000012 07E0          BSET D3,-(A0)
00000014 00FC          DC.W $FC
00000016 07E2          BSET D3,-(A2)
```

```
Src\Dest  Dn An (An) (An)+ -(An) d(An) d(A
   Dn     X  X
   An     X  X
  (An)    X  X
  (An)+   X  X
  -(An)   X  X
  d(An)   X  X
```

Multiple editor tasks can be started and data transfer between windows is supported. It should be noted that the editor's window refreshing appears a bit sluggish. In fact, Abacus' documentation warns you can hang up the system if you cause too many refreshes too quickly.

Error-search functions are provided in the editor. If the error-file switch is enabled during a compile, errors are written to a file that can be used by the editor in these functions. This feature comes in very handy if there are more than a few errors in the source code. If you know in advance there won't be many errors, you may want to turn off the error-file switch and use the correct error feature of the assembler. This feature is discussed later.

AssemPro's editor does not support macros, but a "string" of commands can be attached to each function key. A command is comprised of a two or three letter abbreviation indicating the desired action. These commands can be appended (up to 120 characters) and assigned to one of the function keys. Unfortunately, AssemPro does not provide a command for every function the editor supports. Most of the commands are related to cursor movement, block operations, or search and replace.

The tables, or help, task is utilized best in the editor. If the cursor is placed on an assembly instruction and the Help key is pressed, the tables window is brought to the foreground with an explanation of that instruction's syntax in the window. In addition to 68000 instructions, tables containing the syntax and return values of Amiga operating system calls are provided. It is possible to create your own tables to help with your programming.

After a program is written, it goes to the assembler. The assembler is a two-pass compiler and provides a number of menu options that can be

(continued)

used when compiling your programs. One extremely useful option is its ability to compile from the source currently loaded in the editor. This is an excellent feature demonstrating the interactive nature of the product. When switching back and forth from the editor to the assembler, load times are effectively eliminated.

A previously mentioned option lets you fix errors detected during assembly. Of course, not all errors can be repaired with this option. When an error is detected, the assembler stops and displays the source code in error. If the error is not correctable, the only choice you have is to cancel the compile and return to the editor. If AssemPro can continue when you correct the error, you will be given the opportunity to retype the line. Another "plus" is the ability to apply the correction to the original file. This means you do not have to make the correction twice.

The last major utility provided with AssemPro is the debugger. The debugger, like the other utilities, is menu driven and provides a number of useful features. You can select symbolic disassembly, hex versus ASCII data display, and hex versus decimal number display. Functions are provided that let you search for data, perform calculations, define breakpoints and single-step execution.

The debugger also provides two more useful functions called the disassembler and the reassembler. These can be used to output what you see in the debugger to a file or other device, such as the printer. The difference between the disassembler and the reassembler is their output formats. The reassembler's only purpose is to output source text that can be assembled again by the assembler. The disassembler, on the other hand, provides formatting options for the output text.

I should also mention a set of "libraries" and include files are provided that will assist in interfacing with the operating system. Complete assembly language programs can be written using the editor, assembler, debugger and libraries. Nothing else is required to make assembly programs—not even a linker. The link step is done automatically by the assembler. Thus, Abacus can legitimately claim that AssemPro is "the complete assembler language development package for the Amiga."

Personal Notes

Unfortunately, a "legitimate claim" does not always mean the product is perfect for everyone. It would have been unfair to you, and to Abacus, if I ignored all the strengths and goals of AssemPro just because it does not follow my personal programming practices. Therefore, I have separated the review of the product from my reservations. Please note that my reservations are just that—MY reservations. I don't pretend to speak for every programmer, but some may find the following comments about AssemPro important.

The most significant reservation I have is, I can't link my assembly modules with object modules written in other languages. A modest statement in the documentation passes this problem off as something that can be solved by writing a conversion program. I didn't have that in mind and I will leave that to someone else.

Several programming practices of mine were rejected by the compiler. I referred to my 68000 programming manuals, and although most of these techniques were used in the programming examples, I could not find sufficient documented evidence to call them bugs. The practices in question are these:

1. With most assemblers, beginning a line with an asterisk indicates a comment, not a label. AssemPro generates an error if you do this.

2. You cannot put a label on a line by itself without a colon. It can be followed by anything except emptiness.

3. Local labels must be defined with a leading backslash instead of the customary trailing dollar sign ($). (AssemPro's local labels can be more than a single digit.)

4. AssemPro does not support XREF or XDEF. This may be because there is no need for a linker.

I found these problems while trying to assemble one of my existing programs. Each of these can be overcome. But it will not be a trivial task if you have a large program.

My final reservation concerns the documentation. A few pages should have been included covering the concepts of assembly programming on the Amiga. Additionally, better examples and more in depth explanations of the functions should have been included. Finally, the error message chapter could be improved tremendously. When I got error "(-257) End of file," I looked it up in the book. It said "(-257) End of file."

Although I do have a few reservations about AssemPro, it is a good product. Obviously, a lot of time and thought have gone into AssemPro's development. The interactive nature of the utilities provides an excellent programming environment, and with a few minor revisions in future releases of the product, it will become an assembler for everyone.

•AC•

AssemPro
List $99.95

Abacus Software
5370 52nd Street
Grand Rapids, MI
(616) 698-0330

APL.68000

A programming language of power and elegance comes into its own on the Amiga.

by Roger Nelson

Contrary to widespread belief, APL is easy to use and highly efficient for certain kinds of programming. APL (acronym for A Programming Language) is presently seen as somewhat exotic, and is not widely used on microcomputers. It is inherently very powerful, and is superb for fast and flexible program development. My Amiga provides remarkable feature approximations usually available only on large computers. But one facility I missed was an APL language. Consequently, I was quite excited when I heard of the APL.68000 package for the Amiga, developed by MicroAPL and distributed by the Spencer Organization.

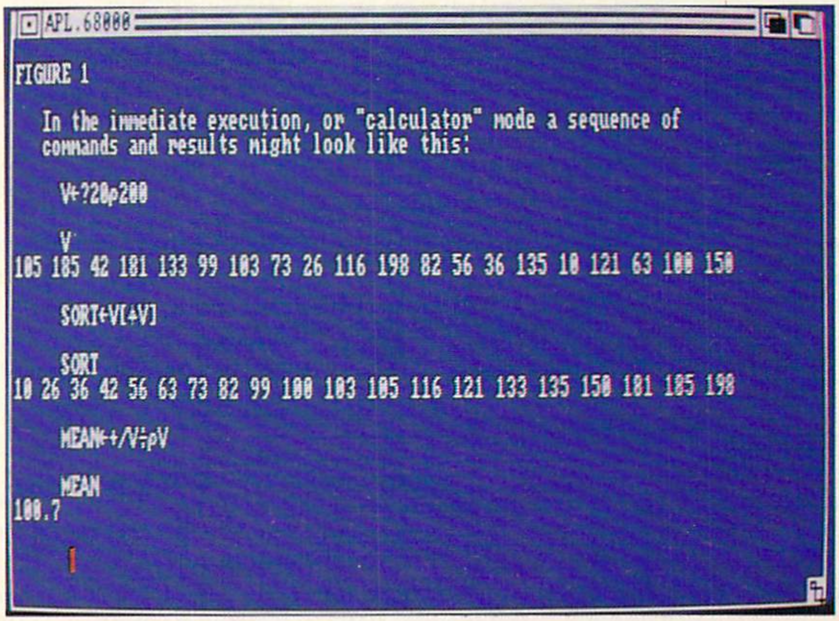
APL.68000 exceeded my expectations. It has virtually total compatibility with the IBM mainframe standard, VSAPL, and is a more refined implementation than other available microcomputer APL packages. System integration and operation are smoother than most new releases. Most important, they didn't cut corners on this package. This is a full APL language, with the capacity to do almost anything a mainframe version can do, albeit somewhat slower. It works fine with 512K of memory, but would really hum with a megabyte or more. Since the 68000 microprocessor can directly address far more memory, the Amiga with APL has almost unlimited potential.

The Language

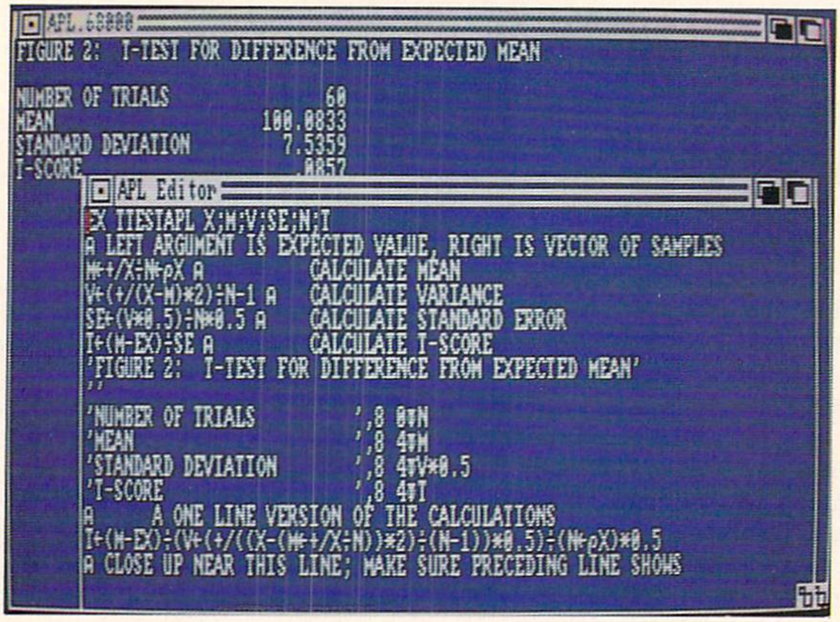
To effectively describe this software package, it is necessary to say a little about the APL language. It is not yet widely used outside the large computer environment. But the development of high quality implementations for microcomputers, especially those with user-friendly interfaces, will probably soon result in an increasing user base. APL originated as a concise mathematical language. As such, it is inherently oriented more to the expressive requirements of the human than to the hardware needs of the computer. It is a collection of powerful primitive functions and operators (like the set of mathematical symbols), but is much more extensive. These are the commands of the language, implemented in machine code, and used by the programmer in an immediate execution mode or in user defined functions. The latter are the equivalent of programs, and are interpreted, not compiled, making development and testing simple and fast.

One feature that clearly distinguishes APL is that all data structures, scalars, vectors, and multi-dimensional arrays are treated alike and require no special declarations. Variables may be local or global, and the latter are immediately accessible to all functions or operations within the workspace. Tables of numbers can be added or multiplied as easily as individual items; indexing and selection are accomplished with a few keystrokes, and APL requires none of the loops or counts other languages use for such operations. Manipulations for characters and numbers are completely analogous, and equally powerful. In APL, both the built-in and user-defined functions are used dynamically, with the output from each function forming the input for the next. A single line of code may be the equivalent to a dozen lines or more in BASIC or C.

(continued)

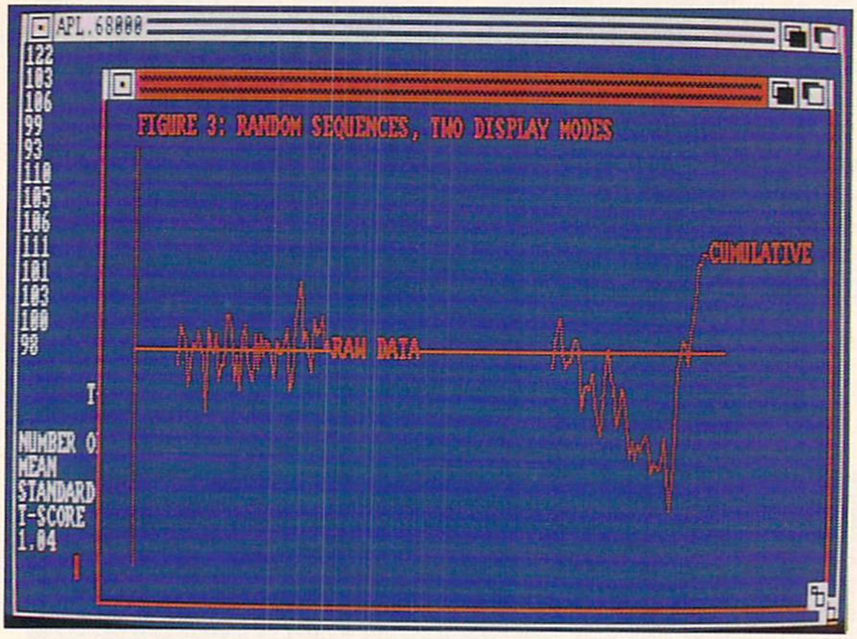


```
FIGURE 1

  In the immediate execution, or "calculator" mode a sequence of
  commands and results might look like this:

    V←?20⍴200

    V
105 185 42 181 133 99 103 73 26 116 198 82 56 36 135 10 121 63 100 150

    SORT←V[⍋V]

    SORT
10 26 36 42 56 63 73 82 99 100 103 105 116 121 133 135 150 181 185 198

    MEAN←+/V÷⍴V

    MEAN
100.7
```



```
FIGURE 2:  T-TEST FOR DIFFERENCE FROM EXPECTED MEAN

NUMBER OF TRIALS              60
MEAN                    100.0833
STANDARD DEVIATION        7.5359
T-SCORE                    .0857
```

```
EX TTESTAPL X;M;V;SE;N;T
⍝ LEFT ARGUMENT IS EXPECTED VALUE, RIGHT IS VECTOR OF SAMPLES
M←+/X÷N←⍴X ⍝          CALCULATE MEAN
V←(+/(X-M)*2)÷N-1 ⍝   CALCULATE VARIANCE
SE←(V*0.5)÷N*0.5 ⍝    CALCULATE STANDARD ERROR
T←(M-EX)÷SE ⍝         CALCULATE T-SCORE
'FIGURE 2:  T-TEST FOR DIFFERENCE FROM EXPECTED MEAN'
''
'NUMBER OF TRIALS          ',8 0⍕N
'MEAN                      ',8 4⍕M
'STANDARD DEVIATION        ',8 4⍕V*0.5
'T-SCORE                   ',8 4⍕T
⍝     A ONE LINE VERSION OF THE CALCULATIONS
T←(M-EX)÷(V←(+/((X-(M←+/X÷N))*2)÷(N-1))*0.5)÷(N←⍴X)*0.5
⍝ CLOSE UP NEAR THIS LINE; MAKE SURE PRECEDING LINE SHOWS
```



```
122
103
106
99
93
110
105
106
111
101
103
100
98

NUMBER O
MEAN
STANDARD
T-SCORE
1.04
```

The language is modular in that a defined function may stand alone, or may be called in another function. Modules can be tested and corrected separately, then easily combined. Although many programmers believe APL is difficult (mostly because the compact code uses special characters to symbolize its large variety of functions), those new to programming will find it easy to write simple but surprisingly powerful programs. (Experienced programmers usually have to overcome serious tendencies to iterate.)

Documentation

My first actual experience was with the software in a Beta test form, and only the on-line documentation—a rather severe test of the user friendliness of the package. But APL.68000 takes advantage of the Amiga Intuition interface so well that it is easy to get started. Double-clicking the APL disk icon produces a window of workspace icons, including one that opens a clear workspace for new material. The rest of the workspaces have descriptive names, such as AMGRAPH, AMFILE, and AMTOOLS. Opening each reveals a set of functions for Amiga graphics, file management, mouse parameters, etc. One might expect these, or at least hope for a library of defined functions to be used in programs along with the primitives that make up the APL language.

The documentation provided by MicroAPL for the Commodore Amiga system is very good. There is a well done 60 page booklet detailing the special Amiga facilities, and a 300 page, spiral bound manual for the APL language. The second section, "Getting Started," is a model of clear, sensitively paced introductory lessons which integrate straight information with well posed examples for hands-on practice. Another section covers "APL Concepts," defining the workspace, explaining data structures, describing built-in functions and operators, and developing the theme of user defined functions.

The last and largest part of the manual is the "Reference" section. The formal and detailed definition of APL.68000 is found in the reference section. The index and table of contents also seem competent. The worst complaint I can make is that throughout the language manual, overstruck characters are badly printed and hard to read. But the content is understandable and effective. In short, I think most people could quickly learn to write productive code using these materials.

Workspaces

The functions and variables used for a particular project or task in APL are typically stored together in a workspace that has the character of a subdirectory or a large file. APL.68000 saves workspaces with an icon, and the workspaces can be accessed with either mouse or keyboard. Functions and variables can be copied easily from one workspace to another, allowing the development of extensive and flexible libraries of program modules. The Amiga package comes with eight such libraries, for accessing graphics, sound, files, menus, requesters, windows, I/O, and miscellaneous tools.

I will describe AMGRAPH as an example. This workspace has functions for pen movement and line drawing, and a POLYLINE function that draws a series of lines connecting any number of points defined by a matrix of X, Y coordinates. Other functions draw and fill rectangles, polygons, arcs, and ovals, and there is flexible control of colors and linetypes, as well as fill patterns, which are controllable to the level of individual pixels. Labels or text in a variety of styles can be inserted anywhere in the field. These well designed low-level graphics functions provide effective direct access to the Amiga's color graphics capabilities (but not, apparently, to the sprites and bobs used in animations). Combined with the APL language, they constitute a remarkably facile development environment for graphics applications. The default screen for drawing is the original APL.68000 window, but another workspace, WINDOWS, contains functions that allow creation of user defined windows and new logical screens. Here again, the result is flexible control over resolution (up to 640 by 400 pixels), color (up to 32 colors), and window placement and dimensions.

The SOUND workspace is also very interesting and easy to use, with a similarly broad spectrum of functions tailored to the Amiga's capabilities. Without going into detail, I can say all the libraries of Amiga specific functions are thoughtfully done. The result is the Amiga and APL are mutually enhanced.

A demonstration workspace gives a general introduction, presented by an APL function that randomly selects a sequence of descriptive text and graphic modules. While the workspaces previously described contain only locked functions (I hope this is changed in later releases) many functions are open in the DEMO workspace. Because they have excellent internal documentation, they can be used as models for user programs. Such models are very helpful, especially for new users, since they act as a "picture worth a thousand words." APL's compact nature can be forbidding, but I think this workspace is a good beginning for an effective on-line introduction. With a broader spectrum of open functions performing both simple and sophisticated tasks, it could serve as an important source of education and encouragement for newcomers to the language.

Multitasking

APL.68000 utilizes multitasking in a number of ways. For example, the VT100 style terminal emulator (which was easy to set up and worked perfectly) can be called under program control or from the menu bar. In either case, it operates as a task, and coexists happily with concurrent APL or Workbench activities. Although a facility for downloading mainframe functions and variables does not appear to be provided in the emulator, it is possible to capture material by copying the screen representation from the terminal emulator window via the clipboard to a concurrent window or workspace. This process also works for uploading, since the screen representation is automatically translated by the emulator. The emulator has both APL and ASCII modes, and they are easily switchable. I was able to work with no difficulty in both the CMS and the APL environments of an IBM 3081.

Multitasking makes APL on the Amiga even more flexible than the usual mainframe versions; several concurrent APL sessions may be generated. For example, the system function for printing text or graphics may be started in a window, then moved to background. The same interpreter is shared by separate APL tasks, conserving memory, and the tasks may be run either in a foreground/background mode or in several screen windows. Input can be made only to the active or foreground window, but processing and output to windows continues for all tasks.

Menus and Features

The Amiga Intuition Interface is used naturally and effectively to manage APL functions and features. In most cases, it provides an alternative to keyboard commands. The PROJECT menu allows mouse and requester control for loading, saving, copying and dropping workspaces, and listing functions and variables. The EDIT menu manages the full-screen editor as a separate task. You can have multiple functions open, making it easy to copy from one to another or to the clipboard. The editor is handled either with the mouse or with keyboard equivalents, using Amiga or control keys. Its response to various programming and typing mistakes seems generally graceful, and informative requesters are used to report errors and warnings. In addition, APL's standard "del" editor is always available, and works in the usual line-oriented manner, enhanced with cursor movement and editing keys.

An INTERRUPT menu provides Break, Suspend, and Resume functions. The terminal emulator can be accessed from the TERMINAL menu. Finally, a SPECIAL menu allows opening a new APL session, changing workspace size, and a choice of operation in an expert mode, where warning requesters (e.g., "Existing FNS/VARS will be overwritten") are suppressed. Functions are also provided (in the MENUS workspace) for creating your own menus, and (in the DIALOG workspace) for setting up requesters within your own APL functions. This means these powerful features of the Amiga system are available to the APL programmer, and that developers can capitalize on both the APL language and the

(continued)

Table 1: Time for various operations, in seconds

	APL.68000	AmigaBASIC
Disk Access	12.6	31.4
Floating Point	7.5	16.1
Trig Function	87.7	40.8
String Extraction	95.6	27.5
Sorting	0.08	39.6
Random Generation	7.7	2.7
Screen Printing	38.2	88.9
Looping	49.7	3.6

Amiga's operating system. As a bonus for software developers, a run-time system is included with APL.68000, making it possible to create packages that are completely transparent to the end-user.

Considerable flexibility is provided by the globally accessible Amiga clipboard, which is available from APL and may be used to transfer text either between APL tasks or to any other window, by cutting and pasting. But my favorite feature is an even more immediate "session manager." It allows text selection on the screen (e.g., the previous command), using cursor movement or the mouse, for editing and resubmission as a command to APL. Having to retype commands in AmigaDOS has always struck me as unfortunate. So this is a special bonus; it is something that should be imitated in any operating system. I think the computer should always be used for repetitious stuff, leaving the operator free for problem solving and creative work. The designers of APL.68000 for the Amiga appear to have a similar philosophy.

Performance

I did several simple Benchmark tests comparing APL.68000 and AmigaBASIC. In some cases the differences are striking, but of course APL is a different type of language (e.g., it operates directly on arrays, without loops), and some of the results reflect style differences rather than practical comparisons of speed in actual programs. A disk access test opened a file, wrote 1000 lines of 46 characters, closed the file, reopened it and read back the 46,000 characters to memory. The floating point test did 10,000 multiplications and stored them in an array, and the trigonometric test did the same for 10,000 sine calculations. A string test used a loop that extracted a substring 10,000 times. A BASIC ripple sort was compared with APL's "grade-up" primitive function, and the generation of 1000 random numbers, with and without screen printing was compared. Finally, a loop with no processing was repeated 10,000 times. The results (in seconds) are shown in Table 1.

These results indicate that APL.68000 interfaces well with the Amiga, with both disk access and screen printing more than twice as fast as in AmigaBASIC. The math results are mixed, but it is worth noting that APL.68000 uses larger real numbers for such calculations, which may have slowed the repeated trig function. When looping was required for the APL function, as in the string extraction, BASIC was more than 3 times as fast. The plain loop test shows part of the reason: 10,000 loops in BASIC requires 3.6 seconds; in APL 49.7! The most startling difference was in the time for sorting 100 random numbers. BASIC took 39.6 seconds, while the APL function was so fast it was hard to measure—an indication of the power provided by APL's large number of primitive functions and operators.

On another level, the overall philosophy of APL is oriented not so much to speed as to efficiency. A few examples will demonstrate the efficiency and simplicity of APL programming. In Figure 1, the variable V is created by drawing 20 random numbers from the set 1 to 200. V is displayed in its original order, then SORT is created with the grade up primitive (this is the process used in the comparison test above). Finally, the mean of V is calculated using only six keystrokes. Figure 2 shows the APL editor with a user defined function that computes and displays a statistical test and related values. The last line presents (just as a curiosity) the same calculations programmed in the infamous one-line style that once characterized APL. Figure 3 shows the output of an easily readable eight-line function that creates a new window and uses a few items from the graphics library to display and label data from a white noise simulator.

Conclusions

For anyone who already knows APL, this software is a splendid enhancement for the Amiga, which is capable of turning APL into a surprisingly effective substitute for an IBM mainframe running VSAPL. For others, I believe the combination of Amiga's Intuition with APL.68000 may remove the barriers and diffuse the myths that have prevented APL from reaching a large audience.

The distributor is making a commitment to education, and has provided the APL.68000 package to some 300 high schools around the country. The price is $99, including the runtime version, making this package not just affordable, but a terrific value. There is currently a program of substantial discounts for students and educators. Versions are also available for Macintosh, Atari ST, and as a 68000 coprocessor board for the IBM PC/XT/AT. This could mean a rapid growth of interest, and good applications software, especially since the Spencer Organization will support developers of runtime applications in APL.

•AC•

APL.68000 $99.00
Requires 512K

Spencer Organization, Inc.
366 Kinderkamack Road
P.O. Box 248
Westwood, New Jersey 07675
(201) 666-6011

by John Steiner

Bug Bytes

The Bugs & Upgrades Column

Recently, I bought a hard disk for my A2000 and discovered many Amiga packages lack hard drive support. Some programs cannot work on a hard disk at all. There are a few commercial programs that demand to be in the ROOT directory. The VIP Professional spreadsheet is one; Digi-Paint is another. It works from a subdirectory on a hard disk, but does not load its title screen and does not save pictures with icons, unless it resides in the ROOT dir. When writing software, developers should minimize special startup sequences. The list assign commands in my startup sequence is nearly a page long.

MiceoSearch is shipping City Desk version 1.1. The desktop publishing program now accepts files created with Scribble! and WordPerfect. A separate clip art disk of 25 pictures is included also. The upgrade is free to version 1.0 users. For details contact:

MicroSearch
9896 Southwest Freeway
Houston, TX 77074
(713) 988-2818

The Director lets you create scripts which can be put together into animated video sequences. Reportedly, if the files have been modified by a program other than Deluxe Paint, the Director gurus on certain IFF files. The solution is to load them into Deluxe Paint and re-save.

There's also something funny related to variables. Make sure they are more than one letter long. According to the report, "I used q,r,s,t for something, and the "s" and "t" variables had problems ... I changed them to 'urx' and 'ulx' and life was fine...."

The Right Answers Group
P.O. Box 3699
Torrance, CA 90510
(213) 325-1311

There is a new hard disk driver program for the expansion drawer on the A2000. If you have read error problems on a hard disk when the Amiga is in overscan mode, you need the new driver. It has been posted to all the major communications links, or you may get it from your dealer for free. You won't need to do anything critical, just copy the new program to the expansion drawer.

Publisher Plus is a replacement and upgrade for Publisher 1000. Many bugs have been fixed. A couple of new features have also been added to make page-creation easier. Improvements include selecting a directory to work on. When changing directories, Publisher 1000 locks into that directory for good. This problem has been fixed with Publisher Plus, and the directory requestor now includes "select" gadgets for df0:, df1:, and dh0:. Publisher Plus has a much faster TYPE mode when editing text within a guide. Publisher 1000 was really slow in this mode. The program also allows you to Lock and Unlock elements on the page. When they are locked, you can't move them, so putting layers of guides in place becomes a much simpler process. If you have Publisher 1000, send your original disk and $18.50 ($15.00 + $3.50 shipping and handling) to:

Northeast Software Group
Attn: Publisher Plus Upgrade
165 Dyerville Ave
Johnston, RI 02919
(401) 273-1001

If you are using the QuarterBack hard disk backup program, check your version number carefully—there is a bug in version 1.1. When restoring from backups, version 1.1 mismatches the directory pointers and the files to which they point. Version 1.3 corrects that problem. According to a spokesperson at Central Coast Software, they have sent recall notices to all distributors, and have replaced all version 1.1 packages in retail and wholesale outlets. They have only received registration cards from about 1/3 of the packages sold, and they would like to notify any 1.1 version owners of the free upgrade. To upgrade to the latest version, send your original disk to Central Coast Software with legible instructions as to where to send the replacement disk. The upgrade is free.

(continued)

Central Coast Software
Attn: Quarterback Upgrade
268 Bowie Drive
Los Osos, CA 93402
(805) 528-4906

Sublogic's Jet version 1.0 has several major problems. First, Jet 1.0 doesn't work correctly with expansion memory. To determine which version you have, look in the bottom right corner of the main menu selection screen. Secondly, the brackets that highlight SAM missile silos in the combined scenario are not in CHIP RAM with expansion memory, causing the display to show them as garbage characters. Next, the bombing-only scenarios do not work. Any missiles or bombs released just turn and fly away toward nothing in particular.

Contact Sublogic, and you can get version 1.01, which fixes both bugs. The brackets look like brackets when using expansion memory. Note, all missiles and bombs released in the bombing-only scenarios fly properly from the front of the Jet with the update.

SubLOGIC Corporation
Attn: Upgrades
P.O. Box 4019
Champaign, IL 61820
(217) 359-8482

According to a press release from Electronic Arts of San Mateo, CA, dated February 26, 1988, "Effective April 1, 1988 all new, original Creativity titles published by Electronic Arts will be free of copy-protection." The company also announced that it will remove copy protection from all existing Deluxe products (including Deluxe Paint II, Deluxe Music Construction Set, Deluxe Video 1.2 and Deluxe Print).

If you have purchased an existing copy-protected Deluxe product, Electronic Arts will replace the program free of charge within ninety days of the purchase date (proof of purchase required). Thereafter, a standard $7.50 replacement charge applies. Additionally, customers who have taken advantage of Electronic Arts backup offer ($20.00 for an uncopy-protected backup copy), or upgrade offer ($57.00 for an uncopy-protected upgrade) will receive their order and a $20.00 gift certificate good for the purchase of the following creativity titles direct through Electronic Arts: Deluxe Music Construction Set, Deluxe Paint II, Instant Music, Deluxe Productions, Deluxe Video 1.2, Intellitype and the Deluxe Library Art & Music Data Disks.

Great idea, Electronic Arts!!

Electronic Arts
1820 Gateway Drive
San Mateo, CA 94404
(415) 571-7171 x379

David Britt of Madison, MS reports an apparent leap year bug in CLImate. No matter what date your system clock shows, if it is a leap year, and you are printing an ASCII file and putting the date on each page, CLImate prints a date one day earlier. In other words, if the system clock has March 10, 1988, CLImate prints out March 9, 1988. David tested this for other dates and years, but only leap years showed the problem. I tested my version of CLIMate, the latest available, and it has the same problem.

I also found WordPerfect exhibits exactly the same symptoms with its date function, implying that the problem might be a bug in the Amiga system date function. On the other hand, David tested Maxiplan, and I tested Analyze!, both of which handled the date correctly. I have also used several other programs that use the date function that work properly.

Professional Page from Gold Disk has a problem that took several hours for us to track down. After developing a newsletter for our local users group, I tried to print it. The document would not print on our QMS laser printer. I printed it to disk, and tried to copy it to the printer—still no luck. After trying to print each page separately, pages two, three and four printed fine, but page one would not print. We selectively removed articles from the page, and finally determined which article was causing the problem.

By deleting one paragraph at a time, we traced the problem to the paragraph that wouldn't print. In that paragraph was an underlined phrase. We deleted the underline and the document printed. As it happened, the underlined phrase also was hyphenated, so we shut off hyphenation, re-underlined the phrase and it printed again. The obvious workaround here is to be sure to turn off hyphenation for any phrase that must be underlined. Thanks to Jacques Chatenay for spending much time working on this problem.

If you have any workarounds or bugs to report, or if you know of any upgrades to commercial software, you may notify me by writing to:

John Steiner
c/o Amazing Computing
P.O. Box 869
Fall River, MA 02722
(617) 678-4200

...or leave EMail to Publisher on People Link or 73075,1735 on CompuServe.

•AC•

Hardware Edition

Amazing COMPUTING™

Your Original AMIGA™ Monthly Resource

AMIGA PRODUCT GUIDE

Accelerators
Hard Drives
Memory Expansion
Midi Interfaces
Video
& More Categories

MicroBotics means Amiga™-Power!

Whichever Amiga you own –or plan to buy– we have the expansion you need

For the Amiga 2000...

HardFrame/2000

Super Speed DMA SCSI Interface

If your application calls for super-speed uninterrupted access to your harddisk, HardFrame/2000 is your answer. This is a high-end, no holds barred SCSI interface that operates at bus speeds. With cable pinouts designed for compatibility with low cost Macintosh hard drives, one HardFrame/2000 can support up to seven devices. Word-length data transfer, FIFO buffering, true DMA, all mounted on a metal frame suitable for mounting standard SCSI 3.5" drives "hard-card" style (or, if you prefer, cable connected to a bay mounted or external disk). Available March/April. Suggested List price $329.

SB2000 Adaptor Card

StarBoard2 Portability

Large numbers of MicroBotics StarBoard2 owners have moved over to the A2000. To protect their investment in our technology we've made available a simple, low-cost adaptor card that permits the installation of a "de-cased" StarBoard2 inside the Amiga 2000 (in the first 100-pin slot). When adapted to the 2000, StarBoard2 is still fully functional autoconfiguring memory plus you get access to all the StarBoard2 MultiFunction options– StickyDisk, Math chip, parity or the new SCSI Module. Available now. Suggested list price is only $49.95.

8-UP! FastRAM

Maximum Memory in One Slot!

The FastRAM card that every Amiga owner will eventually come to –why limit yourself to the possibility of only two megabytes per slot when 8-UP! will take you all the way to the top of the autoconfiguration memory space of EIGHT MEGABYTES ! 8-UP! uses an exclusive MicroBotics-designed memory module, PopSimm, that frees the user to install his own, conventional DIP-style DRAM in standard SIMM sockets on 8-UP!. If you use 256k PopSimms you can install two megabytes on 8-UP!; if you install 1 meg PopSimms, you can install eight megabytes on one card! In either case you can install the memory chips yourself for maximum flexibility and mininum cost. 8-UP! will also accept conventional SIMM memory. 8-UP! is a power efficient, zero wait state, autoconfiguring design. 8-UP! will be available 2nd quarter of 88. Suggested list prices start at $199.

For the Amiga 500...

M501 Memory+Clock

Half a Meg at a Great Price!

As we are all coming to realize, a 1-megabyte Amiga (at least) is a necessity not an option. When you add the inboard 512k memory and clock module to your A500, make it a MicroBotics-brand, plug compatible work-alike. It uses the exact same kind of memory and the exact same clock and battery. And note that just like Commodore and unlike some third-party expansions, we use a long-lived rechargeable Ni-cad battery by Varta– which you'll never have to replace! Set the MicroBotics clock using the same software (on your WorkBench disk) as you use for the Commodore clock. What's the difference? You get to keep $21 compared to the Commodore version. M501 has a suggested list price of only $179.

MicroBotics, Inc.

Great Products Since the Amiga Was Born!

811 Alpha Drive, Suite 335
Richardson, Texas 75081
(214) 437-5330

SOLD ONLY THROUGH YOUR AMIGA DEALER
Tell your dealer he can quick-order from MicroBotics directly – no minimum quantity –show him this ad!

StarBoard2/500

Two Megs and a Choice of Modules

The premier memory expansion for the A1000 is now available on the A500. In a sleek, redesigned case with an independent power supply strong enough to power StarBoard2 and another A1000-style StarBoard2, all the power and flexibility of this great expansion device is available to you. Up to 2 megabytes of autoconfiguring, zero-wait state FastRAM, MultiFunction or SCSI module capability for either math chip/StickyDisk functions or fast SCSI harddisk interfacing. StarBoard2/500 also has a unique LED diagnostic/confidence light to indicate the powered up state of your Amiga and your expansion memory. Another A1000 style StarBoard2 can be connected to the expansion bus pass-UP (it exits through the top of the case) for a total of FOUR megabytes of memory and two modules. Suggested list price $339 and up.

For the Amiga 1000...

StarBoard2

The Expansion Product of Choice

The superb memory expansion for the Amiga 1000, still going strong! Up to 2 megabytes of zero-wait state, autoconfiguring FastRAM in a sleek, all steel Amiga-colored case plus the capability to accept either one of two daughterboard modules: the original MultiFunction Module or the brand new SCSI Module. StarBoard2 is powered by the bus (up to two StarBoard2's can be supported by the A1000) and passes it on. Available now; suggested list price $339 and up.

MultiFunction Module

High Tech at Low Cost

This "daughterboard" installs on any StarBoard2 (all three Amiga models). It features a socket and software to support the Motorola 68881 Math Chip as an I/O device (MicroBotics pioneered this approach on the Amiga –now directly supported in the math libraries in the new AmigaDOS1.3). StickyDisk gives you the most "bullet-proof" rebootable ram disk –its hardware write protection turns the whole device into a solid state, superspeed disk. alternately, parity checking of StarBoard2 memory can be enabled when extra parity RAM is installed. Finally, the MultiFunction Module carries an easy to use battery-backed clock to set your system time on start-up. Available now; suggested list price $90.00.

StarDrive Module

Speedy, Low-cost SCSI Interface

As an alternative to the MultiFunction Module, all models of StarBoard2 can accept this new hard disk interface. StarDrive affords you cost-effective, pseudo-DMA access to Macintosh compatible SCSI drives and other third-party SCSI devices. Fast, easy to install including driver software and disk diagnostics. StarDrive also has a battery backed clock to set your system time on boot-up. Available now. Suggested list price: $129.95

MouseTime

The Port Saving Clock

The easiest-to-use, most cost effective implementation of a battery-backed mouse port clock for the A1000. MouseTime passes the port through for joysticks or other devices. Complete with WorkBench software. Available now. Suggested list of $39.95.

"Amiga" is a registered trademark of Commodore-Amiga. "StarBoard2", StarBoard2/500", "HardFrame/2000", "8-UP!", "PopSimm", "StarDrive", and "MouseTime" are trade names of MicroBotics products.

Amiga Product Guide:

Hardware Edition

Accelerators/Co-Processors 58
Clocks 58
Memory Expansion 58
Hard Drives 60
Other Media 63
Expansion Chassis 67
SCSI Controllers 67
Interfaces 68
Networking 68
Printer Drivers 69
Audio 69
MIDI Interfaces 69
Video 69
Graphics 70
Modems 70
Disk Drives 72
Amiga Modifications 72
Miscellaneous 72
Vendor List 73

Amazing Computing™ provides this guide as a compilation of products from all companies who have responded to our multiple requests for information. The descriptions have been adapted directly from the information supplied by developers. Accordingly, the listings are not intended as reviews, but rather as information for active users in the ever-growing Amiga market. AC™ will not be held responsible for omissions or errors (including prices). If you detect any errors or omissions, please let us know in writing, as soon as possible--Ed.

Amiga - Commodore Computers Users Show

Don't Miss the most Exciting Show on the West Coast Exclusively Devoted to the Amiga and Commodore Computers*

Santa Clara Convention Center
5001 Great American Parkway
(Next to Great America Park)
Santa Clara, California
Sat., May 14, '88 10-8 PM
Sun., May 15, '88 10-5 PM

One Day Adm. Exh. Only	$10.00
One Day Adm. w/ Lectures	$15.00
Two Day Adm. Exh. Only	$15.00
Two Day Adm. w/ Lectures	$25.00

This Show will Feature:

- Animation & CAD
- Business & Database Software
- Desktop Publishing
- Games & Entertainment
- Graphics
- Hard Drives
- Memory Expansion
- Music Software
- Programming Languages
- Public Domain Software
- Spreadsheets
- Simulators
- Telecom. & Utilities

There will also be two full days of lectures and seminars

This show is a marketplace for buyers and sellers of Amiga - Commodore Computers

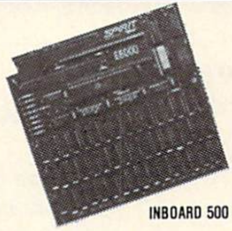
Outside CA (800) 344-3773

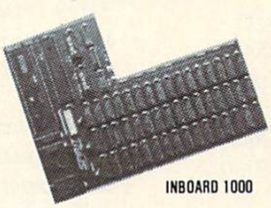
For exhibitor and general information call or write:
Golden Gate Shows • PO Box 587 • Corte Madera, CA 94925 • (415) 388-8893

* Amiga and Commodore are registered trademarks of Commodore-Business Machines Inc.

INNER SPIRIT

NEW LOW PRICE!


INBOARD 500


INBOARD 1000

PROVEN Internal Memory Expansion For The Amiga® 500 and 1000. ONLY $239.50!

Add up to 1.5 MB of Zero Wait State Fast Memory for a total of 2.5 MB° on your Amiga 500 and 2.0 MB on your Amiga 1000.

- Available in various user-expandable configurations from 0K to 1.5 MB with diagnostics included.
- Battery Backed Clock/Calendar standard on 1000 Version, optional on 500 Version.
- Fast, solderless, no-modification installation.
- Safely uses Amiga power in both U.S. and European models. (650 ma. max. 5vdc)
- Detailed installation instructions included.
- Complete set of system utilities (auto-config, ram on/off, readclock, protected ram disk, more) with automatic install routines included.
- Factory technical support.
- Affordable! Only $239.50 for 0K version.
- Satisfaction GUARANTEED, or your money back!

DISK OFFER!

Send $3.00 for a Demo Disk that shows just how easy it is to install your Spirit Memory Expander. Disk is packed with many useful PD programs and utilities.

Specity A-500 or A-1000

°Includes the A-501 memory card

See your dealer or contact Spirit Technology.

220 West 2950 South
Salt Lake City, Utah 84115
(801) 485-4233
FAX (801) 485-6957
Major Credit Cards
1-800-433-7572

SPIRIT™ TECHNOLOGY

SPIRIT TECHNOLOGY CORPORATION

Accelerators/ Co-Processors

Mach II $300
CPU accelerator for the A2000. Nearly doubles speed. Hardware and software compatible. Does not require special memory.
Available May 1988
Aminetics, P.O. Box 982-205, Whittier, CA 90608
(213)-698-6170

PC-Elevator 386 $1795
A2000 compatible accelerator board with 16MHz 80386 processor and zero wait state memory on a local 32-bit bus.
Applied Reasoning Corp., 86 Sherman St., Cambridge, MA 02140
(617) 492-0700

68020/68881 CPU $1,495
68020/68881 or 68882 CPU board for the A2000, or 1000 with expansion chassis. 14MHz 68020, up to 25MHz 68881/68882.
CSA

Over 030 $495
68030 platform board fits CSA 68020 CPU BD or piggyback BDS. Includes software analysis circuitry.
CSA

68020/68881 or
68882 Piggyback BDS $895
Piggyback boards for A500, A1000 for math intensive programs. Installed in 68000 socket inside Amiga.
CSA, 7564 Trade St., San Diego, CA 92121
(619) 566-3911

Bridgecard w./5.25 $699.95
8088 co-processor board for A2000. Multitask with MS-DOS in the background.
Commodore Business Machines, 1200 Wilson Drive, West Chester, PA 19380
(215) 431-9100

Processor Accelerator $149.95
$199.95 with Math Co-Processor socket
Runs 68000 instructions at double speed. Internal installation. Optional math co-processor socket.
Available 2nd quarter 1988
Creative Microsystems, Inc., 10110 S.W. Nimbus #B-1, Portland, OR 97223
(503) 684-9300

32 bit Memory Expansion Board
unpopulated $595;
2Mb $1267; 4Mb $1939
Up to 4Mb of 32-bit dynamic RAM can be connected to the Hurricane for even faster performance.
Finally Technologies

A2000 K-card Adapter $100
Adapts Hurricane to A2000
Finally Technologies

Hurricane 68030
Interface System $990
Finally Technologies

Hurricane Accelerator
16 MHz, $995; 20MHz., $1095
Combines a Motorola 68020 microprocessor and 68881 math coprocessor on a board replacing the 68000 microprocessor in A1000, A500, A2000.
Finally Technologies, 25 Van Ness Ave., San Francisco, CA 94102
(415) 564-5903; FAX (415) 626-4455

MultiFunction Module $99.95;
$295 with 68881
Daughterboard for the StarBoard2. Includes Motorola 68881 Math Chip as an I/O device support and StickyDisk.
MicroBotics, Inc., 811 Alpha Drive, Suite 335, Richardson, TX 75081
(214) 437-5330

Clocks

A-Time $39.95
Battery-backed clock for the A1000.
Akron System Development, P.O. Box 6408, Beaumont, TX 77705
(409) 833-2686

TimeSaver/1000 $79.95
Multifunction battery-backed real-time clock saves RAM memory, connects via the keyboard port.
C Ltd.

TimeSaver II/1000 $99.95
Includes the original features of TimeSaver/1000, plus auto key repeat, enhanced command line editing functions and more.
C Ltd.

TimeSaver II/2000 $79.95
Multifunction device provides hardware-based macro key definitions, battery-backed clock and more.
C Ltd.

TimeSaver II/500 $79.95
Multifunction device provides hardware-based macro key definitions, battery-backed real-time clock and more.
C Ltd.

TimeSaver/1000-
2000 Upgrade Kit $29.95
Upgrades the TimeSaver/1000 for use with the A2000. Includes adaptor cables, new operating system, ROM.
C Ltd, 723 East Skinner, Wichita, KS 67211
(316) 276-6322; FAX (316)-267-0111

MouseTime for A1000 $39.95
Battery-backed mouse port clock passes the port through for joysticks and other devices.
MicroBotics, Inc., 811 Alpha Drive, Suite 335, Richardson, TX 75081
(214) 437-5330

Memory Expansion

2M $599 with 2Mb installed;
$399 with 0Mb
A full Zorro I memory board. Can contain up to 2Mb of memory, includes ASDG Recoverable RAM Disk.
ASDG, Inc.

2MI $399
Full Zorro II memory board can contain up to 2Mb of memory. Can also be used with a A1000 and A500 with card rack, includes ASDG Recoverable RAM Disk.
ASDG, Inc.

8M With 8Mb installed, $2599;
6Mb, $2099; 4Mb, $1549;
2Mb, $999; 0Mb, $499
A full Zorro I memory board. Can contain up to 8Mb of memory, includes ASDG Recoverable RAM Disk.
ASDG, Inc.

8MI With 8Mb, $2500; 6Mb, $2099
; 4Mb, $1549; 2Mb, $999; 0Mb $499
Full Zorro II memory board. Completely autoconfiguring, operates with no wait states. Includes ASDG Recoverable RAM Disk.
ASDG, Inc.

Mini-Rack-D $325
Full Zorro I card rack which can contain two Zorro I expansion cards. Contains 85 watt power supply. Cutouts in back for cables.
ASDG, Inc.

Mini-Rack-E $499
Full Zorro I card rack can contain two Zorro I expansion cards and one 3.5" hard disk drive. Includes internal power supply.
Available June 1988
ASDG Inc., 925 Stewart St., Madison, WI 53713
(608) 273-6585

Alegra 512K, $299; 2Mb, $699
Compact memory expansion unit for the A1000.
Access Associates, 491 Aldo Avenue, Santa Clara, CA 95054-2303
(408) 727-0256

Amiga 2Mb Expansion for 2000/1000/500 $500
2Mb memory expansion. Includes pass-through. Different configurations available for each Amiga model
Alphanetics, P.O. 339, Forestville, CA 95436
707-887-7237

Byte Box
unpopulated, $299.95;
512K, $399.95; 1Mb, $499.95;
2Mb, $699.95
Expands the Amiga's memory to 2Mb. Has own power supply.
Byte by Byte, Aboretum Plaza II, 9442 Capitol of Texas H-way N., Suite 150, Austin, TX 78759
(512) 343-4357

aMEGA Board $499.95
Fully populated 1Mb RAM expansion for the A1000 features auto configuration with AmigaDOS 1.2 and Buss Pass-Thru.
C Ltd.

aMEGA Board-U $299.95
Unpopulated 1Mb RAM expansion for the A1000. Features standard auto configuration with AmigaDOS 1.2 and Buss Pass-Thru.
C Ltd.

RAM/Clock $59.95 with 0k;
$179.95 with 512K
512K memory expansion for the A500 features four-layer circuit board for better noise immunity. Direct plug-in replacement for A-501 memory expansion.
C Ltd., 723 East Skinner, Wichita, KS 67211
(316)-267-0111

32 Bit RAM $595+
Up to 2Mb static 32 bit 14MHz RAM or up to 32Mb 32 bit dynamic RAM.
CSA, 7564 Trade St., San Diego, CA 92121
(619) 566-3911

A2052 2Mb RAM Expander $499.95
2Mb RAM expansion for A2000 available with 512K or 1Mb.
Commodore Business Machines

A501 512K Expansion $199.95
Additional 512K of RAM for A500. Includes clock.
Commodore Business Machines, 1200 Wilson Drive, West Chester, PA 19380
(215) 431-9100

2mega1 N/A

0 to 2Mb RAM board for the A1000
Computer Expansion Products

2mega2 N/A

0 to 2Mb RAM board for the A2000.
Computer Expansion Products

2mega5 N/A

0 to 2Mb RAM board for the A500.
Computer Expansion Products

512K5 N/A

0 to 512K RAM board with clock for the A500.
Computer Expansion Products

8mega1 N/A

2 to 8Mb RAM board for the A1000.
Computer Expansion Products

8mega2 N/A

O to 8Mb RAM board for the A2000.
Computer Expansion Products, Inc., 3596 South 300 West, #10, Salt Lake City, UT 84115 (801) 264-8238

AX 1000/AX 2000 RAM Board $749+
RAM board for the A1000, available in 1 and 2 Mb versions.
Comspec Communications

RE 2000 $225
RAM board for the A2000. With 0k.
Comspec Communications, 74 Wingold Ave., Toronto, Ont., Canada M6B 1P5
(416) 785-3553/FAX (416) 785-3668

M-C 500 512K RAM Board with Clock $149.95
Internally located, easy to install board adds 512K bytes to the A500. Includes battery-backed real-time clock with calendar.
Creative Microsystems, Inc., 10110 S.W. Nimbus #B-1, Portland, OR 97223
(503) 684-9300

RC4 RAMcard N/A
Expandable memory expansion with sockets for up to 4Mb of 256k x 4 DIP DRAMS. 0k installed.
Digitronics, P. O. Box 579, Hatfield, PA 19440
(215) 361-1991

AUTOBOOT/ FAST RAM Expansion Cards
512K, $245; 1Mb, $445; 2Mb, various
Optional RAM modules with AUTO-BOOT for IMPACT A500-HD/RAM subsystem. Also available with 2Mb.
Great Valley Products Inc., P.O. Box 391, Malvern, PA 19355
(800) 426-8957

Ammeg I 0k board, $153.95
A1000 1Mb external memory expansion. Zero wait state design, auto-configuration under 1.2. Fully populated board also available.
Kline-Tronics, 10 Carlisle Ct., York, PA 17404
(717) 764-4205

Insider for A1000 $399.95
Adds 1Mb to the Amiga 1000. Features fast RAM, clock/calendar, ten-year battery. Fits inside the Amiga case.
Michigan Software, 43345 Grand River, Novi, MI 48050
(313) 348-4477

(continued)

5-Power/2000 $279+
8Mb of memory and I/O for the A2000. Includes two AT-style serial ports, socket and software for 68881 Math chip.
Available Fall 1988
MicroBotics, Inc.

8-UP! Fast RAM $195+
FastRAM card gives you 8Mb of autoconfiguration space. Zero wait state, autoconfiguring design.
Available Fall 1988
MicroBotics, Inc.

M501 Memory and Clock Expansion $159
Inboard 512K memory and clock module for the A500.
MicroBotics, Inc.

SB 2000 Adaptor $49
Adapts the StarBoard2 for the A2000.
MicroBotics, Inc.

StarBoard 2 $349+
Memory expansion for the A1000, up to 2Mb of zero wait state, autoconfiguring FastRAM..
MicroBotics, Inc.

StarBoard 2/500 $349+
Up to 2Mb of autoconfiguring, zero wait state FastRAM for the A500. SCSI module capability for either math chip/StickDisk functions or SCSI hard disk interfacing.
MicroBotics, Inc., 811 Alpha Drive, Suite 335, Richardson, TX 75081
(214) 437-5330

Amiga 1000 Memory Expansion Board $600
2Mb memory expansion card with DRAM. Continuous refresh, pass-through, interconnect card, external power supply.
Micron Technology, Inc.

Amiga 2000 Memory Expansion Board $495
2Mb memory expansion card, populated with DRAM. Continuous refresh, pass-through, interconnect card, external power supply.
Micron Technology, Inc.

Amiga 500 Memory Expansion Board $600
2Mb memory expansion card, populated with DRAM. Continuous refresh, pass-through, interconnect card, external power supply.
Micron Technology, Inc.

Micron Memory Board MB-25-D1-PPS & MB-15-D1-PPS
MB-25-D1-PPS (with 2Mb), $600; MB-15-D1-PPS (with 1Mb), $500
Includes externally mounted chassis, interconnect card, wall-mount power supply (optional), pass-through.
Micron Technology, Inc.

Micron Memory Board MB-25-D2 & MB-15-D2
MB-25-D2 (with 2Mb), $495; MB-15-D2 (with 1Mb), $395
1Mb or 2Mb memory board for the A2000. Mounts directly into the expansion slot.
Micron Technology, Inc.

Micron Memory Board MB-25-D5 & MB-15-D5
MB-25-D5 (with 2Mb), $600; MB-15-D5 (with 1Mb), $500
Memory expansion board for the A500 in externally mounted chassis with interconnect card, power supply, pass-through.
Micron Technology, Inc., 2805 E. Columbia Rd., Boise, ID 83706
(800) MICRON-1; (208) 386-3800

The Advantage + $199
(unpopulated—0K)
2 Mb RAM expansion for use in A2000 or subsystem. Uses inexpensive 256K x 1 DRAM. Expandable up to 6 Mb.
Pacific Peripherals, P.O. Box 14575, Fremont, CA 94539
(415) 651-1905

EXP-1000 $399.95
1Mb RAM expansion for the A500. Autoconfiguring, no jumpers needed. Easy installation.
Progressive Peripherals & Software

EXP-512 $169.95
512K RAM expansion for the A500. Auto configuring. Includes battery-backed clock/calendar.
Progressive Peripherals & Software

EXP-8000+
8Mb RAM expansion for the A500. 2Mb of autoconfiguringRAM on-board with optional 68010 processor or 68881 math coprocessor. Easy installation.
Progressive Peripherals & Software, 464 Kalamath St., Denver, CO 80204
(303) 825-4144; FAX (303) 893-6938

Inboard 500/Inboard 1000
500, $279.50; 1000, $299.50
Internal 1.5Mb unpopulated RAMboard with optional time calendar. Zero wait state, autoconfiguring design. For the A500 and A1000
Spirit Technology Corp.

Inboard Memory Expansion $279.50+
For the A500 and A1000. 4-layer buffered PC board which mounts internally to the 68000 MPU socket. Provides RAM ranging from 0.5 to 2.5 Mb.
Spirit Technology Corp., 220 W. 2950 S., Salt Lake City, UT 84115
(800) 433-7572; (801) 458-4233

Supra Drive RAM Expansion Module $399
1-2Mb RAM upgrade for A500/A1000 Supra Drives or Supra SCSI interfaces. Installs in the interface, powers from hard disk.
Supra Corporation

Supra RAM $149.95
For the A500. Easy-to-install internal 0.5Mb RAM upgrade with clock. Mounts in expansion slot on bottom of case.
Supra Corporation, 1133 Commercial Way, Albany, OR 97321
(503) 967-9075

Hard Drives

AM-1308/1000 $7999.95
Hard disk drive for the A1000 includes SCSI host/controller, 1.308Gb hard disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-1308/2000X $7899.95
Hard disk drive for the A2000 includes SCSI host/controller and 1.308Gb hard disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-1308/500 $7949.95
Hard disk drive for the A500 includes SCSI host/controller, 1.308Gb hard drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-161/1000 $2899.95
Hard disk drive for the A1000 includes SCSI host/controller, 161.02Mb hard disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-161/2000X $2799.95
Hard disk drive for the A2000 includes SCSI host/controller, 161.02Mb hard disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-161/500 $2849.95
Hard disk drive for the A500 includes SCSI host/controller, 161.02Mb hard drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-21/2000I $699.95
Hard disk drive for the A2000 includes SCSI host/controller, 21.18Mb hard disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-24/1000 $999.95
Hard disk drive for the A1000 includes SCSI host/controller, 23.85Mb hard disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-24/2000I $799.95
Hard disk drive for the A2000 includes SCSI host/controller, 23.85Mb hard disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-24/2000X $899.95
Hard disk drive for the A2000 includes SCSI host/controller, 23.85Mb hard disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-24/500 $949.95
Hard disk drive for the A500 includes the SCSI-500 host/controller and a 34Mb hard disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-31/2000I $899.95
Hard disk drive for the A2000 includes SCSI host/controller, 31.77Mb hard disk drive. Can be mounted internally. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-332/1000 $4499.95
Hard disk drive for the A1000 includes SCSI host/controller, 332.04Mb hard disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-332/2000X $4399.95
Hard disk drive for the A2000 includes SCSI host/controller, 332.04Mb hard disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-332/500 $4449.95
Hard disk drive for the A500 includes SCSI host/controller, 332.04Mb hard drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-34/1000 $1249.95
Hard disk drive for the A1000 includes SCSI host/controller, 34.45Mb hard disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-34/2000I $999.95
Hard disk drive f or the A2000 includes SCSI host/controller, 34.45Mb hard disk drive. Can be mounted internally. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-34/2000X $1149.95
Hard disk drive for the A2000 includes SCSI host/controller, 34.45Mb hard disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-34/500 $1199.95
Hard disk drive for the A500 includes SCSI host/controller and 34.45Mb hard drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-51/1000 $1499.95
Hard disk drive for the A1000 includes SCSI host/controller, 51.68Mb hard disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-51/2000I $1299.95
Hard disk drive for the A2000 includes SCSI host/controller, 51.68Mb hard disk drive. Can be mounted internally. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-51/2000X $1399.95
Hard disk drive for the A2000 includes SCSI host/controller, 51.68Mb hard disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-51/500 $1449.95
Hard disk drive for the A500. Includes SCSI host/controller, 51.68Mb hard drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-65/1000 $1699.95
Hard disk drive for the A1000 includes SCSI host/controller, 65.50Mb hard disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-65/2000I $1499.95
Hard disk drive for the A2000 includes SCSI host/controller, 65.50Mb hard disk drive. Can be mounted internally. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-65/2000X $1599.95
Hard disk drive for the A2000 includes SCSI host/controller, 65.50Mb hard disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.*C Ltd.*

(continued)

AM-65/500 $1649.95
Hard disk drive for the A500 includes SCSI host/controller, 65.50Mb hard drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-71/2000I $1999.95
A complete hard disk drive for the A2000 includes SCSI host/controller, 71.65Mb hard disk drive. Can be mounted internally. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-72/2000I $1999.95
Hard disk drive for the A2000 includes SCSI host/controller, 72.06Mb hard disk drive. Can be mounted internally. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-75/1000 $1899.95
Hard disk drive for the A1000 includes SCSI host/controller, 75.35Mb hard disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-75/500 $1849.95
Hard disk drive for the A500 includes SCSI host/controller, 75.35Mb hard drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

SCI-21 $599.95
Add-on hard disk drive subsystem for mounting in the A2000 3.5" internal drive. Requires host/controller, includes 21.18Mb hard disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

SCI-24 $549.95
Hard disk drive subsystem for the A2000 can be mounted in the 5.25" drive bay. Includes 23.85Mb hard disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

SCI-31 $499.95
Hard disk drive subsystem for the A2000 can be mounted in the 3.5" drive bay. Includes 21.77Mb hard disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

SCI-34 $599.95
Hard disk drive subsystem for the A2000 can be mounted in the 5.25" drive bay. Includes 34.45Mb hard disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

SCI-51 $899.95
Hard disk drive subsystem for the A2000 can be mounted in the 5.25" drive bay. Includes 51.68Mb hard disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

SCI-65 $1399.95
Hard disk drive subsystem for the A2000 can be mounted in the 5.25" drive bay. Includes 65.50Mb hard disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

SCI-71 $1899.95
Hard disk drive subsystem for the A2000 can be mounted in the 3.5" drive bay. Includes a 71.65Mb hard disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

SCI-72 $1899.95
Hard disk drive subsystem for the A2000 can be mounted in the 5.25" drive bay. Includes a 72.05Mb hard disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

SCX-1308 $7799.95
External hard drive subsystem includes 1.208Gb hard disk drive, internal power supply. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

SCX-161 $2699.95
External hard drive subsystem includes 161.02Mb hard disk drive, internal power supply, cables and more.
C Ltd.

SCX-24 $799.95
External hard drive subsystem includes 23.85Mb hard disk drive, internal power supply. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

SCX-332 $4299.95
External hard drive subsystem includes 332.04Mb hard disk drive, internal power supply. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

SCX-34 $1049.95
External hard drive subsystem includes 34.45Mb hard disk drive, internal power supply. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

SCX-51 $1299.95
External hard drive subsystem includes 51.68Mb hard disk drive, internal power supply. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

SCX-65 $1499.95
External hard drive subsystem includes 65.50Mb hard disk drive, internal power supply. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd., 723 East Skinner, Wichita, KS 67211
(316) 276-6322; FAX (316)-267-0111

Hard Drive 20Mb+
20Mb with expansion, $695
Hard drives with memory expansion slots. 20Mb and up.
Comp-U-Save, 414 Maple Ave., Westbury, NY 11590
(516) 997-6707/(800) 356-9997

SD 20/40/60 $1399/$1799/$1999
20, 40, 60 Mb hard disk drives.
Comspec Communications, 74 Wingold Ave., Toronto, Ont., Canada M6B 1P5
(416) 785-3553/FAX (416) 785-3668

20Mb Hard Drive $899
Hard drive includes controller card, drive cables, power supply, Seagate ST225 drive and drive cabinet identical to the Escort 2 chassis.
Expansion Technologies

20Mb Hard Drive
(for Escort System 500) $849
For the Escort System 500 owner who wants to purchase a drive unit separately. No cabinet.
Expansion Technologies

40Mb Hard Drive $1399
Includes controller card, drive cables, power supply, Seagate ST251 drive and drive cabinet identical to the Escort 2 chassis.
Expansion Technologies

40Mb Hard Drive (for Escort System 500) $1200
For the Escort System 500 owner who wants to purchase a drive unit separately. No cabinet.
Expansion Technologies

Escort 2 20Mb Hard Drive $999
Includes Escort 2 chassis, controller card, drive cables, power supply, Seagate ST225 drive, drive cabinet, zorro expansion slot.
Expansion Technologies

Escort 2 2Mb Version $599
Escort 2 with two A1000 100-pin zorro slots. One slot contains the 2Mb RAM expansion card. Installs on the Amiga 86-pin bus.
Expansion Technologies

Escort 2 40Mb Hard Drive $1499
Includes Escort 2 chassis, controller card, drive cables, power supply, Seagate ST251 drive, drive cabinet, zorro expansion slot.
Expansion Technologies, 46127 Landing Parkway, Fremont, CA 94538
(415) 656-2890

PHD 20/40 $624/$900
20 or 40Mb hard drive for the A500 and A1000. With SCSI interface, 3.5" form factor, internal power supply.
Phoenix Electronics

PHD-1000 20Mb, $899; 40Mb, $1400; 60Mb, $1500
Hard disk drive features SCSI interface command set, pass-through, more. For the A1000.
Phoenix Electronics, Inc., P.O. Box 156, Clay Center, KS 67432
(913) 632-2159

Hard Drives 20 Mb, $659.95; 40Mb $879.95; 65Mb $949.95
Various-sized hard drives for the Amiga, complete with built-in SCSI controller and fan-cooled hard drive case.
Pioneer Computing, 2469 E. 7000 South #200, Salt Lake City, UT 84121
(801) 942-1174

SS-20 Fixed Drive $995
20Mb fixed drive system includes 20Mb 3.5" Winchester disk drive, interface cable, driver software.
Second Source Systems Inc., 501 Business Parkway, Richardson, TX 75081
(214) 680-8394

Supra Drive for A2000
20Mb, $799; 30Mb, $899; 60Mb, $1599
Hard disk drive includes full DMA support, ROM socket, CLImate, optional SCSI expansion port.
Supra Corporation

Supra Drive for A500 and A1000
20Mb, $799; 30Mb, $995; 45Mb, $1195; 60Mb, $1795; 250Mb, $3995
Hard disk drive includes formatting and utilities software, SCSI expansion port, RAM expansion capability.
Supra Corporation, 1133 Commercial Way, Albany, OR 97321
(503) 967-9075

Other Media

AM-10R/1000 $999.95
Removable media disk drive for the A1000 includes SCSI host/controller, 10.5Mb disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-10R/2000I $799.95
Removable media disk drive for the A2000 includes SCSI host/controller, 10.5Mb disk drive. Can be mounted internally. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-10R/2000X $899.95
Removable media disk drive for the A2000 includes SCSI host/controller, 10.5Mb disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-10R/500 $949.95
Removable media disk drive for the A500. Includes SCSI host/controller and a 10.5Mb disk drive. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

AM-60T/1000 $899.95
Streaming tape back-up system for the A1000 includes SCSI host/controller, 60Mb SCSI tape drive. 30-60Mb capacity tapes available.
C Ltd.

AM-60T/2000I $699.95
Streaming tape back-up for the A2000 includes SCSI host/controller. 60Mb SCSI tape drive can be mounted in the 5.25" drive bay of the A2000. 30-60Mb capacity tapes available.
C Ltd.

AM-60T/2000X $799.95
Streaming tape back-up for the A2000 includes SCSI host/controller, 60Mb SCSI tape drive.
C Ltd.

AM-60T/500 $849.95
Streaming tape back-up system for the A500. Includes SCSI host/controller and 60Mb SCSI tape drive. 30-60Mb capacity tapes available.
C Ltd.

AM-800W/1000 $5999.95
Removable media WORM optical disk drive system for the A1000 includes SCSI host/controller, 801.63Mb optical disk drive.
C Ltd.

AM-800W/2000X $5899.95
Removable media WORM optical disk drive for the A2000 includes SCSI host/controller and 801.63Mb optical disk drive.
C Ltd.

AM-800W/500 $5949.95
Removable media WORM optical disk drive system for the A500. Includes SCSI host/controller and 801.63Mb optical disk drive.
C Ltd.

SCI-10R $699.95
10.5Mb removable media disk drive system to be added to the 5.25" drive bay of an A2000 with a SCSI host. Shipped with cables, 10Mb of PD and commercial demo programs.
C Ltd.

(continued on page 67)

Expanding Reference

Expanding reference is not just an empty promise. The pages of Amazing Computing™ are filled with articles on technical operations and procedures, basic use, and just-plain-fun. The growing library of Amazing Computing's Back Issues contains articles ranging from building your own IBM Disk controller, to setting up your own startup sequence. Amazing Computing™ has repeatedly been the first magazine to offer the Amiga users solid, in depth reviews and hands on articles for their machines.

From the Beginning

Since February 1986, Amazing Computing™ has been providing users with complete information for their Amiga. This store house of programs and information is still available through our back issues. From the Premiere issue to the present, there are insights into the Amiga any user will find useful.

Amazing Computing™ was the first magazine to document CLI.
Amazing Computing™ was the first to show Sidecar™ from COMDEX™ in full detail.
Amazing Computing™ was the first to document a 5 1/4 drive connector.
Amazing Computing™ was the first with a 1 Meg Amiga upgrade hardware project.
Amazing Computing™ was the first magazine to offer serious programming assistance.
Amazing Computing™ was the first to offer Public Domain Software at reasonable rates.
Amazing Computing™ was the first magazine with the user in mind.

Back Issues are $5.00 US, $6.00 Canada and Mexico, $7.00 Foreign Surface

All payments must be made by check or money order in U.S. funds drawn on a U.S. Bank.

Limited Supply

Unfortunately, nothing lasts forever, and the availability of some of our Back Issues is definitely limited. Complete your Amazing Computing™ library today, while these issues are still available by completing the order form in the back of this issue and mail with your check or money order to:

Back Issues
Amazing Computing™
PiM Publications, Inc.
P.O.Box 869
Fall River, MA 02722

Volume 1 Number 1 Premiere 1986

Super Spheres By Kelly Kauffman An ABasic Graphics prog.
Data Virus By J Foust A disease may attack your Amiga!
EZ-Term by Kelly Kauffman An ABasic Terminal program
Miga Mania by P. Kivolowitz Programming fixes & mouse care
Inside CLI by G. Musser a guided insight into the AmigaDos™
CLI Summary by G. Musser Jr. A list of CLI commands
AmigaForum by B. Lubkin Visit Compuserve's Amiga SIG
Commodore Amiga Development Program by D. Hicks
Amiga Products A listing of present and expected products

Volume 1 Number 2 March 1986

Electronic Arts Comes Through A review of software from EA
Inside CLI: part two G. Musser Investigates CLI & ED
A Summary of ED Commands
Live! by Rich Miner A review of the Beta version of Live!
Online and the CTS Fabite 2424 ADH Modem by J. Foust
Superterm V 1.0 By K. Kauffman A term. prog. in Amiga Basic
A Workbench "More" Program by Rick Wirch
Amiga BBS numbers

Volume 1 Number 3 April 1986

Analyze! a review by Ernest Viverios
Reviews of Racter, Barataccas and Mindshadow
Forth! The first of our on-going tutorial
Deluxe Draw!! by R. Wirch An Amiga Basic art program
Amiga Basic, A beginners tutorial
Inside CLI: part 3 by George Musser George gives us PIPE

Volume 1 Number 4 May 1986

SkyFox and Articfox Reviewed
Build your own 5 1/4 Drive Connector By Ernest Viveiros
Amiga Basic Tips by Rich Wirch
Scrimper Part One by P. Kivolowitz prog to print Amiga screen
Microsoft CD ROM Conference by Jim O'Keane
Amiga BBS Numbers

Volume 1 Number 5 1986

The HSI to RGB Conversion Tool
by S. Pietrowicz Color manipulation in BASIC
AmigaNotes by Rick Rae The first of the Amiga music columns
Sidecar A First Look by John Foust A first "under the hood"
John Foust Talks with R. J. Mical at COMDEX™
How does Sidecar affect the Transformer
an interview with Douglas Wyman of Simile
The Commodore Layoffs by J. Foust A look Commodore "cuts"
Scrimper Part Two by Perry Kivolowitz
Marauder reviewed by Rick Wirch
Building Tools by Daniel Kary

Volume 1 Number 6 1986

Temple of Apshai Triology reviewd by Stephen Pietrowicz
The Halley Project: A Mission in our Solar System
reviewed by Stephen Pietrowicz
Flow: reviewed by Erv Bobo
Textcraft Plus a First Look by Joe Lowery
How to start your own Amiga User Group by William Simpson
Amiga User Groups
Mailing List by Kelly Kaufman a basic mail list program
Pointer Image Editor by Stephen Pietrowicz
Scrimper: part three by Perry Kivolowitz
Fun With the Amiga Disk Controller by Thom Sterling
Optimize Your AmigaBasic Programs for Speed by Pietrowicz

Volume 1 Number 7 1986

Aegis Draw: CAD comes to the Amiga by Kelly Adams
Try 3D by Jim Meadows an introduction to 3D graphics
Aegis Images/ Animator: a review by Erv Bobo
Deluxe Video Construction Set reviewed by Joe Lowery
Window requesters in Amiga Basic by Steve Michel
ROT by Colin French a 3D graphics editor
"I C What I Think" Ron Peterson with a few C graphic progs
Your Menu Sir! by B Catley program Amiga Basic menues
IFF Brush to AmigaBasic 'BOB' Basic editor by M Swinger
Linking C Programs with Assembler Routines
on the Amiga by Gerald Hull

Volume 1 Number 8 1986

The University Amiga By G.Gamble
Amiga's inroads at Washington State University
MicroEd a look at a one man army for the Amiga
MicroEd, The Lewis and Clark Expedition reviewed Frizelle
Scribble Version 2.0 a review
Computers in the Classroom by Robert Frizelle
Two for Study by Frizelle Discovery & TheTalking Coloring Book
True Basic reviewed by Brad Grier
Using your printer with the Amiga
Marble Madness reviewed by Stephen Pietrowicz
Using Fonts from AmigaBasic by Tim Jones
Screen SaVer by P. Kivolowitz A monitor protection prog. in C
Lattice MAKE Utility reviewed by Scott P. Evernden
A Tale of Three EMACS by Steve Poling
.bmap File Reader in Amiga Basic by T Jones

Volume 1 Number 9 1986

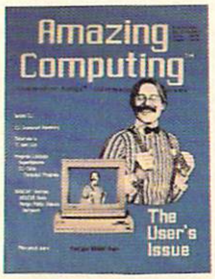
Instant Music Reviewed by Steve Pietrowicz
Mindwalker Reviewed by Richard Knepper
The Alegra Memory Board Reviewed by Rich Wirch
TxEd Reviewed by Jan and Cliff Kent
Amazing Directory A guide to the sources and resources
Amiga Developers A listing of Suppliers and Developers
Public Domain Catalog A listing of Amicus and Fred Fish PDS
Dos 2 Dos review R. Knepper
Transfer files from PC/MS-DOS and AmigaBasic
MaxiPlan review by Richard Knepper The Amiga Spreadsheet
Gizmoz by reviewed by Peter Wayner Amiga extras!
The Loan Information Program by Brian Catley
basic prog. to for your financial options
Starting Your Own Amiga Related Business by W. Simpson
Keep Track of Your Business Usage for Taxes by J. Kummer
The Absoft Amiga Fortran Compiler reviewed by R A Reale
Using Fonts from AmigaBasic, Part Two by Tim Jones
68000 Macros on the Amiga by G. Hull Advance your ability.
TDI Modla-2 Amiga Compiler review by S Faiwisze

Volume 2 Number 1 1987

What Digi-View is... Or, What Genlock Should Be! by J. Foust
AmigaBasic Default Colors by Bryan Catley
AmigaBasic Titles by Bryan Catley
A Public Domain Modula-2 System reviewed by Warren Block
One Drive Compile by Douglas Lovell
Using Lattice C with a single drive system
A Megabyte Without Megabucks by Chris Irving
An Internal Megabyte upgrade
Digi-View reviewed by Ed Jakober
Defender of the Crown reviewed by Keith Conforti
Leader Board reviewed by Chuck Raudonis
Roundhill Computer System's PANEL reviewed by Ray Lance
Digi-Paint....by New Tek previewed by John Foust
Deluxe Paint II ....from Electronic Arts previewed by J. Foust

Volume 2 Number 2 1987

The Modem by Josph L. Rothman efforts of a BBS Sysop
MacroModem reviewed by Stephen R. Pietrowicz
GEMINI or "It takes two to Tango" by Jim Meadows
Gaming between machines
BBS-PC! reviewed by Stephen R. Pietrowicz
The Trouble with Xmodem by Joseph L. Rothman
The ACO Project....Graphic Teleconferencing on the Amiga
by S. R. Pietrowicz
Flight Simulator II....A Cros Country Tutorial by John Rafferty
A Disk Librarian in AmigaBASIC by John Kennan
Creating and Using Amiga Workbench Icons by C. Hansel
AmigaDOS version 1.2 by Clifford Kent
The Amazing MIDI Interface build your own by Richard Rae
AmigaDOS Operating System Calls and
Disk File Management by D. Haynie
Working with the Workbench by Louis A. Mamakos Prog in C

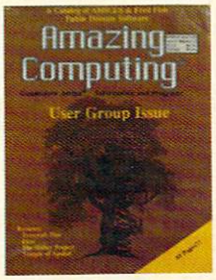

VOLUME 1.1

VOLUME 1.2

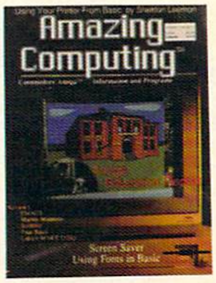
VOLUME 1.3

VOLUME 1.4

VOLUME 1.5


VOLUME 1.6

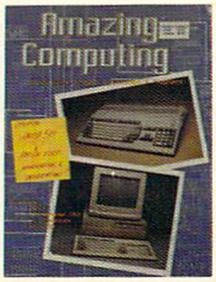
VOLUME 1.7

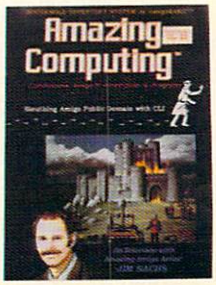

VOLUME 1.8

VOLUME 1.9

VOLUME 2.1

VOLUME 2.2

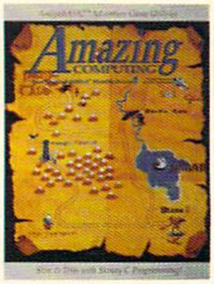

VOLUME 2.3

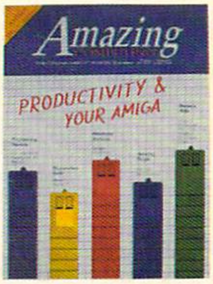

VOLUME 2.4

VOLUME 2.5

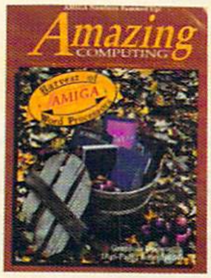
VOLUME 2.6

VOLUME 2.7

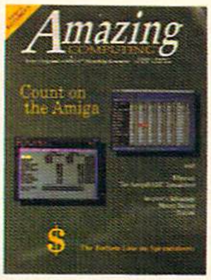

VOLUME 2.8

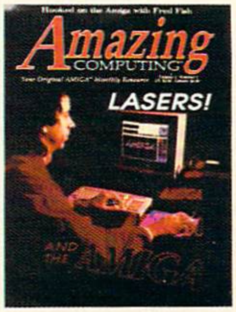

VOLUME 2.9

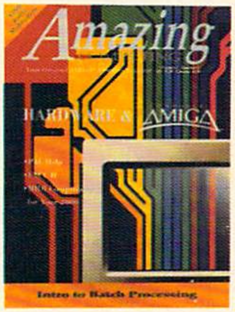
VOLUME 2.10


VOLUME 2.11

VOLUME 2.12


VOLUME 3.1


VOLUME 3.2


VOLUME 3.3

VOLUME 3.4

Volume 2 Number 3

The Amiga 2000™ by J Foust
A First look at the new, high end Amiga™
The Amiga 500™ by John Foust
A look at the new, low priced Amiga
An Analysis of the New Amiga PCs by J. Foust
Speculation on the New Amigas
Gemini Part II by Jim Meadows
The concluding article on two-player games
Subscripts and Superscripts in AmigaBASIC by Ivan C. Smith
The Winter Consumer Electronics Show by John Foust
AmigaTrix by W. Block Amiga™ shortcuts
Intuition Gadgets by Harriet Maybeck Tolly
A journey through gadget-land, using C
Shanghai reviewed by Keith M. Conforti
Chessmaster 2000 & Chessmate reviewedby Edwin V. Apel, Jr.
Zing! from Meridian Software reviewed by Ed Bercovitz
Forth! by Jon Bryan Get stereo sound into your Forth programs.
Assembly Language on the Amiga™ by Chris Martin
Roomers by theBandito Genlocks are finally shipping, & MORE!!!
AmigaNotes by R. Rae Hum Busters... "No stereo? Y not?...
The AMICUS Network by J. Foust
CES, user group issues and Amiga Expo"

Volume 2 Number 4 1987

Amazing Interviews Jim Sachs by S. Hull Amiga Artist
The Mouse That Got Restored by Jerry Hull and Bob Rhode
Sluething Public Domain Disks with CLI by John Foust
Highlights: the San Francisco Commodore Show by S Hull
Speaker Sessions: San Francisco Commodore Show H Tolly
Household Inventory System in AmigaBASIC™ by B Catley
Secrets of Screen Dumps by Natkun Okun
Using Function Keys with MicroEmacs by Greg Douglas
Amigatrix II by Warren Block More Amiga shortcuts
Basic Gadgets by Brian Catley Create gadget functions
Gridiron reviewed by K. Conforti Real football for the Amiga
Star Fleet I Version 2.1 reviewed by J. Tracy Amigain Space
The TIC reviewed by J. Foust Battery powered Clock Calendar
Metascope review by H. Tolly An easy-to-use debugger

Volume 2 Number 5 1987

The Perfect Sound Digitizer review by R. Battle
The Future Sound Digitizer by W. Block Applied Vision's SD
Forth! by J. Bryancomparing JForth amd Multi-Forth.
Basic Input by B. Catley AmigaBASIC input routine for use in
all your programs.
Writing a SoundScape Module in C by T. Fay Programming with MIDI, Amiga and SoundScape by SoundScape author.
Programming in 68000 Assembly Language by C. Martin
Continuing with Counters & Addressing Modes.
Using FutureSound with AmigaBASIC by J. Meadows
AmigaBASIC Programming utility with real, digitized STEREO
AmigaNotes Rich Rae reviews SoundScape Sound Sampler.
More AmigaNotes by R. Rae A furtherlook at Perfect Sound .
Waveform Workshop in AmigaBASIC by J. Shields edit & save
waveform for use in other AmigaBASIC programs.
The Mimetics Pro MIDI Studio by Sullivan, Jeffery
A review of Mimetics' music editor/player.
Intuition Gadgets Part II by H. MaybeckTolly Boolean gadgets
provide the user with an on/off user interface.

Volume 2 Number 6 1987

Forth! by J. Bryan Access resources in the ROM Kernal.
The Amazing Computing Hard Disk Review by J. Foust & S. Leemon In-depth looks at the C Ltd. Hard Drive, Microbotics' MAS-Drive20, Byte by Byte's PAL Jr., Supra's 4x4 Hard Drive and Xebec's 9720H Hard Drive. Also. a look at disk driver software currently under devlopment.

Modula-2 AmigaDOS™ Utilities by S. Faiwiszewsk A
Calls to AmigaDOS and the ROM kernal.
Amiga Expansion Peripheral by J. Foust
Explanaton of Amiga expansion peripherals.
Amiga Technical Support by J. Foust
How and where to get Amiga tech support.
Goodbye Los Gatos by J. Foust Closing Los Gatos.
The Amicus Network by J. Foust West Coast Computer Faire.
Metacomco Shell and Toolkit by J. Foust A review
The Magic Sac by J. Foust Run Mac programs on your Amiga.
What You Should Know Before Choosing an Amiga 1000
Expansion Device by S. Grant
7 Assemblers for the Amiga by G. Hull Choose your assembler
Shakeup Replaces Top Management at Commodore by S. Hull
Peter J. Baczor"by S. Hull Manager at CBM gives an inside look
Logistix A review by Richard Knepper
Organize! by A review Richard Knepper database.
68000 Assembly Language Programming on the Amiga
by Chris Martin
Superbase Personal Relational Database by Ray McCabe
AmigaNotes by Rae, Richard A look at FutureSound
Commodore Shows the Amiga 2000 and 500 at the Boston
Computer Society by H Maybeck Tolly

Volume 2, Number 7 1987

New Breed of Video Products by John Foust...
Very Vivid! by Tim Grantham...
Video and Your Amiga by Oran Sands III
Amigas & Weather Forecasting by Brenden Larson
A-Squared and the Live! Video Digitizer by John Foust.
Aegis Animator Scripts and Cel Animation by John Foust
Quality Video from a Quality Computer by Oran Sands III.
Is IFF Really a Standard? by John Foust..
Amazing Stories and the Amiga™ by John Foust.
All about Printer Drivers by Richard Bielak
Intuition Gadgets by Harriet Maybeck Tolley.
Deluxe Video 1.2 by Bob Eller
Pro Video CG1 by Oran Sands III.
Digi-View 2.0 Digitizer/Software by Jennifer M. Janik
Prism HAM Editor from Impulse by Jennifer M. Janik
Easyl drawing tablet by John Foust..
CSA's Turbo-Amiga Tower by Alfred Aburto
68000 Assembly Language by Chris Martin.

Volume 2, Number 8 1987

This month Amazing Computing™ focuses on entertainment packages for the Amiga. Amazing game reviews...

SDI, Earl Weaver Baseball, Portal, The Surgeon, Little Computer People, Sinbad, StarGlider, King's Quest I,II and III, Faery Tale Adventure, Ultima III, Facets of Adventure, Video Vegas and Bard's Tale.

Plus Amazing monthly columns... Amiga Notes, Roomers, Modula-2, 68000 Assembly Language and The Amicus Network.
Disk-2-Disk by Mathew Leeds
The ColorFonts Standard by John Foust
Skinny C Programs by Robert Riemersma, Jr.
Hidden Messages in Your Amiga™ by John Foust
The Consumer Electronics Show and Comdex.by J Foust

Volume 2 Number 9 1987

Analyze 2.0 reviewed by Kim Schaffer
Impact Business Graphics review by Chuck Raudonis
Microfiche Filer review by Harv Laser
Pagesetter review by Rick Wirch
Gizmoz Productivity Set 2.0 review by Bob Eller
Kickwork review by Harv Laser
Diga Telecommunications Package review by Steve Hull
Mouse Time and Timesaver review by John Foust
Insider Memory Expansion review by James O'Keane
Microbotics Starboard-2 review by S. Faiwiszewski
Leather Goddess of Phobos
reviewed by Harriet Maybeck-Tolly
Lattice C Compiler Version 3.10 reviewed by Gary Sarff
Manx 3.4a Update reviewed by John Foust
AC-BASIC reviewed by Sheldon Leemon
AC-BASIC Compiler an alternative comparison by B Catley
Modula-2 Programming S Faiwiszewski
Raw Console Device Events
Directory Listings Under AmigaDOS by Dave Haynie
AmigaBASIC Patterns by Brian Catley
Programming with Soundscape
Todor Fay manipulate's samples
Bill Volk, Vice-President Aegis Development,
interviewed by Steve Hull
Jim Goodnow, Devekoper of Manx 'C'
interview by Harriet M Tolly

Volume 2 Number 10 1987

Max Headroom and the Amiga by John Foust
Taking the Perfect Screen Shot by Keith Confori
Amiga Artist: Brian Williams by John Foust
Amiga Forum on Compuserve™... Software Publishing
Conference Transcript by Richard Rae
All About Online Conferencing by Richard Rae
dBMAN reviewed by Clifford Kent
Amiga Pascal reviewed by Michael McNeil
AC-BASIC Compiler reviewed by Bryan Catley
Bug Bytes by John Steiner
Amiga Notes by Richard Rae
Roomers by The Bandito
68000 Assembly Language by Chris Martin
The AMICUS Network by John Foust
Amiga Programming:
Amiga BASIC Structures by Steve Michel
Quick and Dirty Bobs by Michael Swinger
Directory Listings Under Amiga-DOS, Part II by Dave Haynie
Fast File I/O with Modula-2 by Steve Faiwiszewski
Window I/O by Read Predmore

Volume 2 Number 11 1987

Word Processors Rundown by Geoff Gamble
ProWrite, Scribble!, and WordPerfect compared
LPD Writer Review by Marion Deland
VizaWrite Review by Harv Laser
Aedit Review by Warren Block
WordPerfect Preview by Harv Laser
Jez San Interview by Ed Bercovitz
The author of StarGlider speaks!
Do-it-yourself Improvements to the Amiga Genlock
Digi-Paint Review by Harv Laser
Sculpt 3D Review by Steve Pietrowicz
Shadowgate Review by Linda Kaplan
TeleGames Review by Michael T. Cabral
Reason Preview: a quick look at an
intense grammar examination application
As I See It by Eddie Churchill Peeking at WordPerfect,
Gizmoz V2.0 and Zing! Keys
Bug Bytes by John Steiner
AmigaNotes by R Rae 4 electronic music books
Modula-2 Programming by Steve Faiwiszewski
devices, I/O, and the serial port
Roomers by The Bandito
68000 Assembly Language by Chris Martin
Chris walks through the display routines
The AMICUS Network by John Foust
Desktop Publishing & Seybold
C Animation Part II by Mike Swinger Animation Objects
BASIC Text by Brian Catley Pixel perfect text positioning
Soundscape Part III by Todor Fay VU Meter and more
Fun with Amiga Numbers by Alan Barnett
File Browser by Bryan Catley
Full Feature BASIC File Browsing utility

Enrich your Amiga
JOIN
an Amiga User Group
TODAY

Volume 2 Number 12 1987

The Ultimate Video Accessory by Larry White
The Sony Connection by Stewart Cobb
15-Puzzle in AmigaBASIC by Zoltan Szepel
Life, Part I: The Beginning by Gerald Hull
The ultra-complex nine bit solution to the "Game of Life."
Amiga Virus! by John Foust
A new Amiga virus has surfaced. Please check your system.
CLI Arguments in C by Paul Castonguay
MIDI Interface Adapter by Barry Massoni
Amiga 1000-style MIDI interfaces can fit A2000s or 500s
Modula-2 by Steve Faiwiszewski
First in a series, a command line calculator in Modula-2.
AmigaNotes by Richard Rae
The audio changes made in the Amiga 500 and 2000.
Animation for C Rookies: Part III by M. Swinger
tackling double-buffering.
The Big Picture by Warren Ring
Amiga™ Assembly language programming for the brave!
Karate Kid Reviewby Stephen R. Pietrlowicz
GO! 64 review by John Foust, James O'Keane, and Rick Wirch
Three C-64 experts investigate a new Amiga 64 emulator.
A-Talk-Plus Review by Brendan Larson
"Full-fledged terminal program" & Tektronics capabilities
Calligrapher Review by John Foust
Animator: Apprentice Review by John Foust
Playing Dynamic Drums on the Amiga by David N. Blank
WordPerfect Review by Steve Hul
Insider/Kwikstart Review by Ernest P. Viveiros Sr
RAM & ROM expansion: Comments and installation tips.
Bug Bytes by John Steiner
Forth! by Jon Bryan
DumpRPort utility for your Multi-Forth toolbox.
As I See It by Eddie Churchill
An offbeat look on Digi-Paint, Portal, and Videoscape 3D.
The Amicus Network by John Foust
The Commodore Show and AmiExpo: New York!

Volume 3 Number 1 1988

AmigaNotes by Richard Rae
Digital music generation on the Amiga .
C Animation Part IV by Michael Swinger
Just when you thought it was safe to go back in the C waters...
Forth by John Bryan
Sorting out CHIP and FAST memory on the Amiga.
The Big Picture by Warren Ring Daring assembler language programming; CLI system calls and manipulating disk files.
Bug Bytes by John Steiner
Roomers by The Bandito
Amiga Dos1.3? 80386-based BridgeBoard for the A2000? More!
As I See It by Eddie Churchill
Opinions, obsevations, & the birth of a new software generation.
68000 ASsembly Language Programming by Chris Martin
"Create a multi-color screen without using Intuition routines!"
Modula-2 Programming by Steve Faiwiszewski
A new contender bursts on to the modula-2 scene!
Amicus Network Special Report: Fall COMDEX by J. Foust
Commodore at COMDEX and new products
The ultimate Video Accessory: Part II by Larry White
Life: Part II by Gerald Hull
"A detailed look at efficient use of the Amiga blitter."
FormatMaster: Professional Disk Formatting Engine byC.Mann
Put Batch language to work on the drudgery of disk formatting .
BSpread by Brian Catley
A full featured AmigaBASIC spreadsheet you can program!
AmigaForum Transcript ed. by Richard Rae
Zoom in on Commodore/ Amiga's Dave Haynie.
Halcalc Review by Chuck Raudonis
"A straightforward, easy to use, functional spreadsheet."
VIP Professional Review by Suzanne Mitchell
Easy stock porfolio management on the Amiga.
Money Mentor Review by Stephen Kemp
A personal finance system beyond your checkbook.
Investor's Advantage Review by Richard Knepper
plus "Poor Man's Guide to the Stock Market."

Volume 3 Number 2 1988

Laser Light Shows with the Amiga by Patrick Murphy
Lasers and the Amiga: A Dazzling Tandem
The Ultimate Video Accessory: PartIII by Larry White
Take the final stepstoward designing your own videos.
Our First Desktop Video by Larry White
Step-by -step guide to organizing & presenting your Amiga video.
Hooked on the Amiga with Fred Fish by Ed Berkovitz
Inside views from the man behind all those "Fish" disks.
Photo Quality Reproduction with the Amiga and Digi-View
by Stephen Lebans
Balancing your Checkbook with WordPerfect Macros by S.Hull
Hand your checkbook worries over to the Amiga.
More Basic Text by Bryan Catley easier text on an Amiga screen
Life: Part III by Gerald Hull
Sries winds up with famed nine-bit calculation & source to LIFER.
Solutions to Linear Algebra through Matrix Computations
by Ropbert Ellis
Simplify matrix algebra with basic operations & routines.
Roomers by Bandito Amiga 3000, Virus news,& Laser Toaster
Bug Bytes by John Steiner
Modula-2 Programming by Steve Faiwiszewski
Catching up with Calc-a source follow-up.
68000 Assembler Language Programming by Chris Martin
Graphics- Part II of Assemgram.
Arazok's Tomb by Kenneth E. Schaefer
"A terrifyig adventure into the world of the occult."
AIRT by Steve Faiwiszewski
An innovative icon-basede programming language.
Forms in Flight by Steve Pietrowicz
Render and Animate objects in 3D!
Silicon Dreams and the Jewel of Darkness by K E. Schaefer
Leisure suit Larry by Kenneth E. Schaefer
Two New Entries From Microbiotics by John Foust
M501 Expansion & Starboard II MultiFunctionboard.
Mindlight 7 and People Meter by John Foust
Phantasie
Ken E. Schaefer with the Amazing Phantasie Character Editor.

Volume 3 Number 3 1988

Take Five! by Steve Hull
Blast your frustrations in these five Amiga games.
Desktop Video,Part IV by Larry White
Put all the pieces together-the desktop video commercial.
The Hidden Power of CLI Batch File Processing
by Joe Rothman
Make your Amiga easier to use with CLI Batch files.
A Conference with Eric Graham edited by John Foust
The mastermind behind Sculpt 3D and Animate 3D.
Perry Kivolowitz Interviewed by Ed Bercovitz
Amiga insights from a major developer and personality.
Jean "Moebius" Giraud Interviewed by Edward L Fadigan
Avant-garde art comes to the Amiga-in dazzling form.
PAL Help by Perry Kivolowitz
All the help you need for a A1000 expansion reliability.
Boolean Function Minimization by Steven M. Hart
A useful digital design tool in AmigaBASIC.
Amiga Serial Port and Midi Compatibility for Your A2000!
by Lynn Ritter and Gary Rentz
Add an A1000-style serial port to your A2000!
Electric Network Solutions the Matrix Way by Robert Ellis
Engineers! Practicle routines for using matrix algebra.
In The Public Domain by C.W. Flatte
Hot flashes and highlights from the latest Fred Fish disks.
The A.M.U.G. BBS List compiled
by Joe Rothman, Chet Solace, and Dorothy Dean 100
514 BBS phone numbers in the U.S. and Canada.
FACC II reviewed by Graham Kinsey
Put a firecracker under your floppy drives.
Uninvited reviewed by Kenneth E. Schaefer
When was the last time a game scared you to death?
Flow reviewed by Pamela Rothman
Turn your brainstorms into mental works of art.
Benchmark Modula-2 Compiler reviewed by Richie Bielak
Program development that beats Pascal to the punch.
Bug Bytes by John Steiner
Stay abreast of today's bugs and tomorrow's upgrades.
Modula-2 Programming by Steve Faiwiszewski
The gameport device and simple sprites in action.
AmigaNotes by Richard Rae
A1000! Create a software-switchable output filter.
Roomers by The Bandito
Inside AmiExpo ... Kickstart 1.4 .. Commodores 'R' Us?
The Big Picture by Warren Ring
Unwiedly arguments to system calls?
Discover the Unified Field Theory!!

Volume 3 Number 4 1988

Highlights from AmiExpo, Los Angeles by Steve Hull
The Amiga shows off its best in the West.
Writing a SoundScape Patch Librarian by Todor Fay
Get your hands dirty working within
the System Exclusive.
Upgrade Your A1000 to A500/2000 Audio Power
by Howard Bassen
Modifications to help your A1000 make sweet music, too!
Amiga Audio Guide
Descriptive listing of all Amiga audio products.
Gels in Multi-Forth by John Bushakra
Push Gels to the limit with these programming tools.
Macrobatics by Patrick J. Horgan Ease the trauma of assembly language programming.
Amiga Audio Sources The folks behind all those audio products.
Take Five! by Steve Hull
Four lightning-paced titles to slash boredom.
Amiga Notes by Rick Rae
Confounded by sound? Take a basic tour of Amiga audio.
The Ultimate Video Accesory, Part V by Larry White
Let's add some flash to our video.
Bug Bytes by John Steiner
The exterminator strikes again.
The Big Picture by Warren Ring
Part II of the eye-opening Unified Field Theory.
Roomers by The Bandito
Hardware hijinx ... Toasted video ... the dream Amiga ... and more!
In the Public Domain by C.W. Flatte
C.W. has hooked the latest Fish disks—here's an inside look.
Time Bandit review by Keith Conforti
A whole video arcade wrapped up in one game!
AudioMaster review by Brendan Larson
Friendly digitizing software that samples in real-time.
Music Mouse review by J Henry Lowengard
Making music without lifting a finger from the mouse.
Amiga-Tax/Canadian Version review by Ed Bercovitz
A Canadian income tax planning, preparation, and
analysis package for the Amiga.
SAM BASIC review by Bryan Catley
A new BASIC which exploits even more unique Amiga features.

To be continued........

To Order Back Issues,

please use the order form on page 127

SCI-60T $599.95

Streaming tape back-up subsystem for the A2000 can be mounted in the 5.25" drive bay. Tape drive requires a SCSI host/controller. 30-60Mb capacity tapes available.

C Ltd.

SCX-10R $799.95

10.5Mb removable media disk drive can be added to any Amiga with a compatible SCSI host/controller. Shipped with cables, 10Mb of PD and commercial demo programs.

C Ltd.

SCX-60T $699.95

Streaming tape back-up subsystem includes a 60Mb SCSI tape drive with an internal power supply. 30-60Mb capacity tapes available.

C Ltd.

SCX-800W $5799.95

800Mb removable media WORM optical disk drive includes an 801.63 Mb drive and uses removable media floppy disks.

C Ltd., 723 East Skinner, Wichita, KS 67211
(316) 276-6322; FAX (316)-267-0111

Halfbac-40, Bacpac $1595+

40Mb streaming tape drive for A2000 or A1000 with expansion chassis. Mounts internally, or externally in Bacpac with tape and hard drive.

CSA

WORM-800 $5995

800Mb optical laser disk drive for A2000 or A1000 with expansion chassis. Use with SCSI interface card and software driver.

CSA, 7564 Trade St., San Diego, CA 92121
(619) 566-3911

BN 1000 $2995

SCSI Bernoulli box for the A1000.

Comspec Communications, 74 Wingold Ave., Toronto, Ont., Canada M6B 1P5
(416) 785-3553/FAX (416) 785-3668

Bernoulli 20Mb Disk Drive

A2000, $1695; A1000, $1795

Fast access time removable disk. Many applications. 20Mb cartridge available, $99.

Inner Connection Inc., 12310 Brandywine Road, Brandywine, MD 20613
(301) 372-8071

Supra Drive FD-10 $1095

10Mb removable floppy disk drive includes formatting and backup software. Reads 360K and 1.2Mb IBM-format disks.

Supra Corporation, 1133 Commercial Way, Albany, OR 97321
(503) 967-9075

Expansion Chassis

ProtoBoard-II $49.95

Prototype board for Zorro-II slots.

Akron System Development, P.O. Box 6408, Beaumont, TX 77705
(409) 833-2686

Highrise $995

Expansion chassis for the A1000. 7 Zorro II slots, 4 IBM slots, CPU slot. Accepts bridge card, several hard drives/tape/WORM, etc.

CSA, 7564 Trade St., San Diego, CA 92121
(619) 566-3911

Escort System 500 $849

Expansion system that includes chassis, two zorro expansion slots, 2Mb RAM card, more. Chassis supports the monitor, leaves room for expansion.

Expansion Technologies, 46127 Landing Parkway, Fremont, CA 94538
(415) 656-2890

Subsystem 1000 $249

3-slot expansion box with pass-through. Uses A2000 form factor cards. 3.5" SCSI hard drive may be mounted inside box.

Pacific Peripherals

Subsystem 500 $249; with drive, $399

2 slot expansion box for the A500 with optional 3.5" floppy drive. Uses A2000 form factor expansion cards.

Pacific Peripherals, P.O. Box 14575, Fremont, CA 94539
(415) 651-1905

SCSI Contollers

ACB-4000 $149.95

SCSI/ST-506 device controller allows any standard ST-506/412 hard drive to be used with any of C Ltd's SCSI Host Controllers.

C Ltd.

ACB-4070 $199.95

SCSI/ST-506 RLL Device Controller allows any RLL certified ST-506/412 hard drive to be used with any of C Ltd's SCSI host/controllers.

C Ltd.

SCSI-1000 $299.95

SCSI controller that utilizes TRUE ANSI standard SCSI protocol and is capable of interfacing with a wide variety of SCSI devices.

C Ltd.

SCSI-2000 $199.95

Zorro-standard device provides an internal SCSI connector and external DB-25 connector. Includes SCSInet 2.03 software.

C Ltd.

SCSI-2506M $369.95

Provides the same features as the SCSI 2000, plus MFM support for two ST-506/412 hard drives.

C Ltd.

SCSI-2506R $399.95

Provides the same features as the SCSI 2000, plus RLL support for two RLL STT-506/412 hard drives.

C Ltd.

SCSI-500 $249.95

SCSI controller features full auto configuration, buss pass-through, three-source power supply option, and SCSInet 2.03 software.

C Ltd., 723 East Skinner, Wichita, KS 67211
(316) 276-6322; FAX (316)-267-0111

A2090 Hard Disk Controller $399.95

Hard disk controller with ST 506 and SCSI capability.

Commodore Business Machines, 1200 Wilson Drive, West Chester, PA 19380
(215) 431-9100

SCSI controller $250

A500, A1000, A2000 compatible SCSI controller, includes software.

Computer Expansion Products, Inc., 3596 South 300 West, #10, Salt Lake City, UT 84115
(801) 264-8238

(continued)

506 Controller $399
Controller card for two ST 506 or IBM drives. Fits in any A1000 card cage and includes card and power supply.
Expansion Technologies, 46127 Landing Parkway, Fremont, CA 94538
(415) 656-2890

IMPACT A2000-SCSI/ RAM Hard Disk Controller
512K, $495; 1Mb, various
AUTOBOOT feature allows the A2000 to boot directly from a hard disk. Also available with 1Mb FAST RAM.
Great Valley Products

IMPACT A500-HD/ RAM subsystem $995
SCSI controller, 20Mb hard disk and FAST RAM expansion add-on subsystem for the A500.
Great Valley Products Inc., P.O. Box 391, Malvern, PA 19355
(800) 426-8957

HardFrame/2000 $329
Hard card-style SCSI interface that operates at bus speeds. Support for up to 7 devices, word-length data transfer, FIFO buffering, true DMA.
Available Spring 1988
MicroBotics, Inc., 811 Alpha Drive, Suite 335, Richardson, TX 75081
(214) 437-5330

Over Drive $249
Hard card style SCSI interface that can be used in the A2000 or expansion chassis for the A1000 and A500. Supports up to 7 devices.
Pacific Peripherals, P.O. Box 14575, Fremont, CA 94539
(415) 651-1905

PCC $180
SCSI hard disk controller card for A1000 and A500.
Phoenix Electronics

PHC-2000
20Mb, $499; 30Mb, $599; 40Mb, $699
Hard card-type SCSI for the A2000.
Phoenix Electronics, Inc., P.O. Box 156, Clay Center, KS 67432
(913) 632-2159

Supra SCSI Interface $199
SCSI interface board allows hard disks to be connected to A500s and A1000s. Includes built-in clock, RAM expansion capability.
Supra Corporation, 1133 Commercial Way, Albany, OR 97321
(503) 967-9075

Interfaces

Amiga GPIB (IEEE-488) $495
General purpose interface bus card follows IEEE-488 specification, allows the Amiga to control up to 14 IEEE-488 devices.
ACDA Corporation

Proto-40K ADC/DAC/DIO Acquisition and Control Board $1,795
Interface board for the A2000 allows full autoconfiguration. Includes DAS library, sample application programs.
ACDA Corporation

Proto-5K $279.95
Analog-to-digital converter data acquisition box for lower-performance applications. Includes range test calibration circuit.
ACDA Corporation, 220 Belle Meade Ave., Setauket NY 11733
(516) 689-7722

AMy-LAB $799
Data acquisition system; plug-in card for the A2000.
Available May 1988
Akron System Development, P.O. Box 6408, Beaumont, TX 77705
(409) 833-2686

Microshare MCS 1050 $135
Amiga-to-IEEE 488 interface.
Comspec Communications

Microshare MCS 6550 $300
256K Centronix printer buffer.
Comspec Communications, 74 Wingold Ave., Toronto, Ont., Canada M6B 1P5
(416) 785-3553/FAX (416) 785-3668

The Fix
RGB, $49.95; Parallel and Serial, $39.95
Compatibility interfaces: RGB Fix; Parallel Fix; Serial Fix.
DATASOUND, 603 Brantley Place, Virginia Beach, VA 23452
(804) 431-1362

Serial Expander 2000 $19.95
Adds an A1000 compatible serial port to the A2000. Lets you use serial devices designed for the A1000 on the A2000.
Golden Hawk Technology, 427-3 Amherst St., Nashua, NH 03063
(603) 424-0269

StarDrive Module $129.95
An alternative to the MultiFunction module. Pseudo-DMA access to Macintosh compatible SCSI drives and other 3rd party SCSI devices.
MicroBotics, Inc., 811 Alpha Drive, Suite 335, Richardson, TX 75081
(214) 437-5330

Access-64 $79.95
Adapter with software allows Amiga to access Commodore 64 and 128 serial drives and printers.
Progressive Peripherals & Software, 464 Kalamath St., Denver, CO 80204
(303) 825-4144; FAX (303) 893-6938

Micro Stuffer $69.95
64K printer buffer works with any computer and printer with a Centronics parallel interface. Includes power supply, 3' output cable for printer, multiple-copy repeat function, clear button, and auto diagnostics.
Supra Corporation, 1133 Commercial Way, Albany OR 97321
(503) 967-9075

Networking

AE2000 Ethernet $899
Ethernet lan controller for the A2000 includes NFS software package. Provides connectivity to many UNIX machines.
Ameristar Technologies, Inc., 47 Whittier Ave., Medford, NY 11763
(516) 698-0834

SCSInet 3.00 $99.95
Local area networking system uses high speed data transfer and multi-user capabilities.
C Ltd., 723 East Skinner, Wichita, KS 67211
(316) 276-6322; FAX (316)-267-0111

Microshare
MTS 8008/MTS 8004 $750/$600
Multi-user printer network.
Comspec Communications, 74 Wingold Ave., Toronto, Ont., Canada M6B 1P5
(416) 785-3553/FAX (416) 785-3668

Printer Drivers

HP PaintJet Interface $50
Preferences driver for the HP-PaintJet.
Lightning Publishing

NEC CP7 Interface $50
Preferences driver for the NEC CP7.
Lightning Publishing

NLQ 342C Interface $50
Preferences driver for the NLQ 324C.
Lightning Publishing

Quadron QuadJet Interface $30
Preferences driver for the Quadram QuadJet
Lightning Publishing, 1821 N. Ohio St., Arlington, VA 22205
(703) 534-8030

Audio

FutureSound $175
Audio digitizer with microphone and editing software.
Applied Visions, Suite 2200, 1 Kendall Sq., Cambridge MA 02139
(617) 494-5417

Micro SMPTE N/A
SMPTE reader allows Music-X to synchronize with video or audio tape decks. Connects via parallel interface, includes pass-through. A1000, A500, A2000 compatible.
MicroIllusions, 17408 Chatsworth St., Granada Hills CA 91344
(818) 360-3715

Perfect Sound $89.95
Sound digitizer records in stereo from any line-level input source. Creates IFF instruments and includes library and software.
Microsearch Inc., 9896 Southwest Freeway, Houston, TX 77074
(713) 988-2818

MIDI Interfaces

ECE MIDI 1000 $64.95
MIDI for A1000 with IN, OUT, THRU and RS-232 bypass.
ECE Research & Development

ECE MIDI 500/2000 $65
MIDI for A500/A2000 with IN, OUT, THRU and RS-232 bypass.
ECE Research & Development, 1651 N. Monroe St, Tallahassee, FL 32303
(904) 681-0786

MIDI Gold 500/2000 $59.95
MIDI with 1 IN, 2 OUTs, 1 OUT/THRU and serial pass-through. No cable required.
Golden Hawk Technology, 427-3 Amherst St., Suite 389, Nashua, NH 03063
(603) 424-0269

Deluxe MIDI Interface $89
MIDI with IN, THRU, two switchable THRU/OUTs and switchable RS-232 pass-through.
Hypertek/Silicon Springs, 120-1140 Austin Ave., Coquitlam, B.C., Canada V3K 3P5
(604) 939-8235

Micro Midi N/A
MIDI with 6 outputs, external clock output, serial pass-through.
MicroIllusions, 17408 Chatsworth St., Granada Hills, CA 91344
(818) 360-3715

MIDI 1000 $49.95
MIDI for A1000 with IN, OUT, THRU.
Mimetics Corporation

MIDI 500/2000 $49.95
MIDI for A500/A2000 with IN, OUT, THRU.
Mimetics Corporation, P.O. Box 1560, Cupertino, CA 95014
(408) 741-0117

Video

LIVE! $295
Digitizes a moving color image in real time from any video source.
A-Squared Distributions Inc., 6114 LaSalle Ave., Oakland, CA 94611
(415) 339-0339

A-Video $49.95
RGB to color composite for A500 and A2000.
Akron System Development, P.O. Box 6408, Beaumont, TX 77705
(409) 833-2686

C-View I $49.95
Allows the RGB video output of any Amiga to drive a standard composite color monitor or the video input of a VCR.
C Ltd.

C-View II $49.95
Allows the Amiga's RGB video output to drive Commodore's 1700 and 1800 series monitors and VHS VCRs that accept chroma/luma inputs.
C Ltd., 723 East Skinner, Wichita, KS, 67211
(316) 276-6322; FAX (316) 267-0111

1084 RGB 13" Monitor $399.95
NTSC composite video encoder with RF modualtor.
Commodore Business Machines, 1200 Wilson Drive, West Chester, PA 19380
(215) 431-9100

GEN/ONE $895.00
Genlock encoder with overlay. Has separate 4C output for super-VHS. Available June 1988
Communications Specialties Inc., 6090 Jericho Turnpike, Commack, NY 11725
(516) 499-0907

V-I 500 $59.95;
$69.95 with RF modulator
Video interface that supports a variety of video standards, provides quality color composite video signal, generates chroma/luma signals.
Creative Microsystems, Inc.

V-I 2000 $59.95;
$69.95 with RF modulator
Same as V-I 500; works on A2000 only.
Creative Microsystems, Inc., 10110 S.W. Nimbus #B-1, Portland, OR 97223
(503) 684-9300

SuperGen $749.95
Genlock and overlay device with many professional features, including RGB encoder.
Digital Creations, 1333 Howe Ave. #208, Sacramento, CA 95825
(916) 344-4825

(continued)

Color Enhancement Filter $19.95
Improves color and contrast while eliminating screen flicker. Improves monitor appearance.
Gladstone Productions

Command Control Console N/A
A switching console designed to interface and control digitizing equipment and two printers.
Available April 1988
Gladstone Productions, 7744 Pickering Ave., Whittier CA 90602
(213) 696-3372

TTL Hi-Res Monitor Interface $99.95
Provides flicker-free interfaced graphic output with three grey scales when used with the Commodore 1901 or Magnavox Monitor 80.
Hypertek/Silicon Springs, 120-1140 Austin Ave., Coquitlam, B.C., Canada V3K 3P5
(604) 939-8235

Impulse Video Digitizer $499.95
Capture, freeze, and digitize any NTSC video source in full frame or single fields.
Impulse, Inc., 6870 Shingle Creek Parkway #112, Minneapolis, MN 55430
(612) 566-0221

MediaPhile $525
Allows computer control of A-V equipment. Edits video tape, records database, plays back automatically from one or two decks.
Interactive Microsystems, P.O. Box 1446, Haverhill, MA O1831
(617) 372-0400

Photon Video Transport Controller N/A
Frame by frame controller allows you manually or automatically move your animation to video tape
Available Spring 88
MicroIllusions, 17408 Chatsworth St., Granada Hills, CA 91344
(818) 360-3715

Digi-Droid $79.95
Motorized filter wheel for Digi-View that allows for faster RGB image catching.
New Tek

Digi-View 3.0 $199.95
Standard video digitizer. Version 3.0 has enhanced image quality, overscan. Supports halfbrite mode.
New Tek

Video Toaster $799.95
High-quality special effects generator. Includes Genlock, frame grabber and DVE.
Available Summer 1988
New Tek, 115 W. Crane St., Topeka, KS 66603
(913) 354-1146

ProGEN $399.95
Professional, software-controllable Genlock for all Amigas.
Progressive Peripherals & Software, 464 Kalamath St., Denver, CO 80204
(303) 825-4144; FAX (303) 893-6938

Perfect Vision $219.95
A video "frame grabber" digitizer with IFF format save. Grab completely still pictures from a VCR or laserdisk.
Sunrize Industries, 3801 Old College Road, Bryan, TX 77801
(409) 846-1311

RM-2 $2995
Sync generator genlock system that overlays graphics and text on video feeds from cameras, film chains, etc. Many features.
Wollner Associates, 3306 Horseman Lane, Falls Church, VA 22042
(703) 533-1236

Graphics

Easyl A500/1000/2000
$399/$449/$499
1024 x 1024 pressure-sensitive digitizer. Universal driver. Includes pegs to facilitate cel animation.
Anakin Research, Inc., 100 Westmore Drive Unit 11c, Rexdale Ont., CANADA M9V 5C3
(416) 744-4246

IS/ONE $495
Graphic digitizing tablet interfaces through many software packages.
Kurta, 3007, Chambers East, Phoenix, AZ 85040
(602) 276-5533

AProCAD Graphics Tablet Package
9 x 6 tablet, $499; 12 x 12 tablet, $599
Up to 1000 lines per inch. Includes 4-button crosshair cursor, Summagraphics MM-Series tablet. Interfaces through the serial port.
R & D L Productions, 11-24 46th Avenue, 2A, Long Island City, NY 11101
(718) 392-4090

IMG Scan II $139.95
Turns your printer into a high resolution image scanner. 256 gray levels, full-color assignable.
Seymor-Radix , P.O. Box 166055, Irving, TX 75016
(214) 255-7490

Modems

Minimodem-A #6216 $99.95
300/1200 baud modem includes speaker with volume control and status indicator LEDs. Powered from the serial port.
APROTEK, 1071-A Avenida Acaso, Camarillo, CA 93010
Orders, (800) 962-5800; information, (805) 987-2454

1200 Baud Modem $149.95
Commodore Business Machines, 1200 Wilson Drive, West Chester, PA 19380
(215) 431-9100

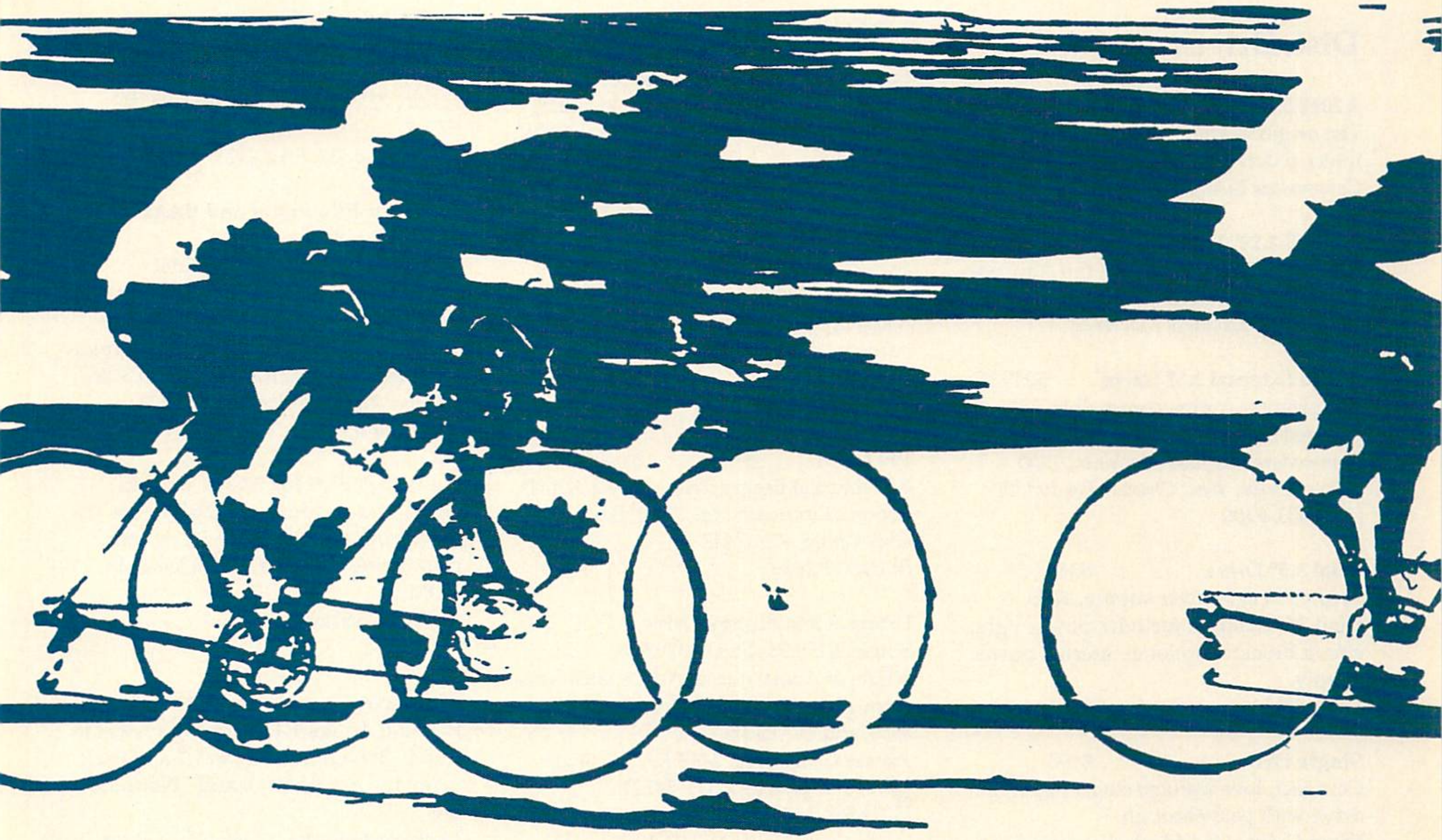
Avatex 1200 E $85.00
Small modem with total Hayes compatibility, CCitt compatibility, call progress detection, and more.
Megatronics, Inc., P. O. Box 3669, Logan, UT 84321
(800) 232-6342

Supra Modem 2400 $179.95
2400 baud modem, compatible with AT commands and common protocol. Features nonvolatile memory, more.
Supra Corporation

Supra Modem 2400AM $219.95
2400 baud modem package. Includes standard Supra Modem 2400 package, telecomm. software, RS-232 cable, A1000 adapter.
Supra Corporation, 1133 Commercial Way, Albany, OR 97321

(503) 967-9075

(continued)



SMARTER. NOT HARDER.

Developing a product for the Amiga market is like a good race; it requires strategy, pacing, and command of your resources. Sometimes, no matter how hard you spin your wheels, you just can't make it to the front of the pack. If the front of the pack is where you want to be, consider placing your advertisement in Amazing Computing.

Amazing Computing has a solid reputation for providing worthwhile and complete information to the Amiga user. An advertisement in Amazing Computing will reach serious Amiga users, knowledgeable buyers—your customers. Whether your marketing goals require a multi-page color spread or a 1/9 page black and white advertisement, AC can help at a price you can afford.

Don't sweat it out with the rest, consider advertising in a smarter market—Amazing Computing. For details contact:

John Fastino, Advertising Manager
PiM Publications Inc.
One Currant Place
Fall River, Ma. 02722
(617) 678-4200

Disk Drives

A2010 3.5" Internal Drive $199.95
The original Amiga compatible 3.5" internal drive.
Commodore Business Machines

A1020T 5.25" External Drive $399.95
5.25" external drive allows the Amiga to utilize 5.25" floppies.
Commodore Business Machines

A1010 External 3.5" Drive $299.95
The original Amiga compatible 3.5" external drive.
Commodore Business Machines, 1200 Wilson Drive, West Chester, PA 19380
(215) 431-9100

Dual 3.5" Drive $345;
with internal power supply, $395
Dual 3.5" external includes power light, circuit breaker, optional internal power supply.
Comp-U-Save

Single Drive $190
Compact, low-wattage single external drive with pass-through.
Comp-U-Save, 414 Maple Ave., Westbury, NY 11590
(516) 997-6707/(800) 356-9997

Floppy Drive
(for Escort System 500) $239
For the Escort System 500. Includes controller card, cables, 23-pin connector that mounts on the rear of the chassis, quiet 3.5" floppy.
Expansion Technologies, 46127 Landing Parkway, Fremont, CA 94538
(415) 656-2890

3.5" Disk Drive
Internal, $159; External, $189
Internal or external 3.5" drives for the Amiga.
HHT Electronics, 2477 N. 450 W., Harrisville, UT 84404
(801) 782-1841

CA-880 3.5"
Floppy Disk Drive $229.95
Small, quiet disk drive supports all Amiga models, formats double-sided, double-density diskettes for up to 880K of storage.
Logical Design Works, Inc., 780 Montague Expwy., #403, San Jose, CA 95131
(408) 435-1445

PFD-135 $2293.5
880K external floppy drive for all Amigas. Features low power drive with pass-through.
Phoenix Electronics

PFD-135I $149.95
3.5" internal floppy drive for the A2000.
Phoenix Electronics, Inc., P.O. Box 156, Clay Center, KS 67432
(913) 632-2159

External 3.5" Floppy Drive
Single, $159.95; Dual $329.95
Single or dual external drives with pass-through for additional drives, metal case, and 18 "cable.
Pioneer Computing, 2469 E. 7000 South #200, Salt Lake City, UT 84121
(801) 942-1174

ProDrive $239.95
External 3.5" disk drive for all Amigas.
Progressive Peripherals & Software

ProDrive 2000 $189.95
Internal 3.5"disk drive for the A2000.
Progressive Peripherals & Software, 464 Kalamath St., Denver, CO 80204
(303) 825-4144; FAX (303) 893-6938

Internal 3.5 Disk Drive
for A2000 $149.95
Faster internal 3.5" drive. Dealer installation recommended.
Studio 7, P.O. Box 4727, Overland Park, KS 66204
(913) 649-0491

Amiga Modifications

People Meter $59.95
Galvanic skin response stress interface. Includes software including several meters, graphics programs, and a game.
Aminetics, P.O. Box 982-205, Whittier, CA 90608
(213) 698-6170

MMI Pal Chip Set $20
Faster, less sensitive alternative PAL chip set for the A1000 daughterboard.
C Ltd., 723 East Skinner, Wichita, KS 67211
(316) 276-6322; FAX (316)-267-0111

Kickstart Eliminator and RAM Expansion Kit
$129.95; $150 installed by CMI
Install ROM into existing locations on the motherboard, gain an additional 256K of fast RAM. Soldering required.
Creative Microsystems, Inc., 10110 S.W. Nimbus #B-1, Portland, OR 97223
(503) 684-9300

Kwikstart Plus for A1000 $169.95
Gives you an additonal 256K to use when running under the 1.2 system, puts the new Amiga 1.2 Kickstart in ROM.
Michigan Software

Multistart for A500 and A2000 $129.95
Puts the Amiga 1.1 operating system in ROM. Switch from 1.2 to 1.1 and back using the Amiga keyboard. No soldering.
Michigan Software, 43345 Grand River, Novi, MI 48050
(313) 348-4477

CPS-500 $99.95
Replacement power supply for the A500. 8 Amps. 3 AC receptacles, transient/spike suppression, more.
Phoenix Electronics, Inc., P.O. Box 156, Clay Center, KS 67432
(913) 632-2159

Miscellaneous

NoRad dB60 Anti-glare/Static/Radiation Filter $129.95
Grounded filter reduces flicker in high-resolution or interlace mode. Easy maintenance.
Brookfield Communications, 3820 Griffith View Drive, Los Angeles, CA
(213) 668-0030, (800) 533-dB60

Profit Enhancement Center
$3599 for 1 POS station with cash drawer; $6999 for 1 POS station and 1 back office station.
Provides all hardware and software for Point of Sale program, inventory control and full accounting.
Clockwork Computers, 4612 Holly Ridge Road, Rockville, MD 20853
(301) 924-5509

X-10 Powerhouse System $45
Allows the Amiga to control lights and small appliances through your existing house wiring.
Compplications, 1727 Parkview, Redlands, CA 92374
(714) 794-5311

ECE Alignment Kit $175
Aligns Amiga 3.5 disk drives. Kit includes alignment disk, program disk, board and cable, manual, more.
ECE Research & Development, 1651 N. Monroe St., Tallahassee, FL 32303
(904) 681-0786

EPYX 500XJ Joystick $19.95
Compact joystick with special grip and trigger-finger firing.
EPYX, 600 Galveston Drive, P.O. Box 8020, Redwood City, CA 94063
(415) 366-0606

Amiga Light Pen $129.95
Hi-res 2-button touch switch pen coupled with transparent driver. Allows user to choose Light Pen, mouse, or both alternately
Inkwell Systems, 5710 Ruffin Road, San Diego CA 92123-1013
(619) 268-8792

MindLight 7 $169
Sound-responsive computer art program lets you visualize music. IFF compatible.
Visual Aural Animation, P.O. Box 4898, Arcata, CA 95521
(707) 822-4800

Amiga Trackball Controller $49.95
2-button trackball controller, compatible with all Amiga mouse software."
Zebra Systems Inc., 78-06 Jamaica Ave., Woodhaven, NY 11421
(718) 296-2385

•AC•

Vendor List

A-Squared Distributions Inc.
6114 LaSalle Ave.
Oakland, CA 94611
(415) 339-0339

Access Associates
491 Aldo Avenue
Santa Clara, CA 95054-2303
(408)-727-0256

ACDA Corporation
220 Belle Meade Ave.
Setauket, NY 11733
(516) 689-7722

Akron System Development
P.O. Box 6408
Beaumont, TX 77705
(409) 833-2686

Alphanetics
P.O. Box 339
Forestville, CA 95436
(707) 887-7237

Ameristar Technologies, Inc.
47 Whittier Ave.
Medford, NY 11763
(516) 698-0834

Aminetics
P.O. Box 982-205
Whittier, CA 90608
(213) 698-6170

Anakin Research, Inc.
100 Westmore Drive Unit 11c
Rexdale, Ontario
CANADA M9V 5C3
(416) 744-4246

Applied Reasoning Corp.
86 Sherman St.
Cambridge, MA 02140
(617) 492-0700

Applied Visions
1 Kendall Sq., Suite 2200
Cambridge, MA 02139
(617) 494-5417

APROTEK
1071-A Avenida Acaso
Camarillo, CA 93010
Orders (800) 962-5800
Information (805) 987-2454

ASDG Inc.
925 Stewart St.
Madison, WI 53713
(608) 273-6585

Brookfield Communications
3820 Griffith View Drive
Los Angeles, CA
(213) 668-0030
(800) 533-dB60

Byte by Byte
Aboretum Plaza II
9442 Capitol of Texas Highway N.
Suite 150
Austin, TX 78759
(512) 343-4357

C Ltd.
723 East Skinner
Wichita, KS 67211
(316) 276-6322
FAX (316) 267-0111

Commodore Business Machines
1200 Wilson Drive
West Chester, PA 19380
(215) 431-9100

Communications Specialties Inc.
6090 Jericho Turnpike
Commack, NY 11725
(516) 499-0907

Comp-U-Save
414 Maple Ave.
Westbury, NY 11590
(516) 997-6707
(800) 356-9997

Compplications
727 Parkview
Redlands, CA 92374
(714) 794-5311

Computer Expansion Products, Inc.
3596 South 300 West, #10
Salt Lake City, UT 84115
(801) 264-8238

Comspec Communications
74 Wingold Ave.
Toronto, Ontario
CANADA M6B 1P5
(416) 785-3553
FAX (416) 785-3668

Creative Microsystems, Inc.
10110 S.W. Nimbus #B-1
Portland, OR 97223
(503) 684-9300

CSA
7564 Trade St.
San Diego, CA 92121
(619) 566-3911

(Vendor List continued)

Digital Creations
1333 Howe Ave. #208
Sacramento, CA 95825
(916) 344-4825

Digitronics
P.O. Box 579
Hatfield, PA 19440
(215) 361-1991

ECE Research & Development
1651 N. Monroe St.
Tallahassee, FL 32303
(904) 681-0786

Expansion Technologies
46127 Landing Parkway
Fremont, CA 94538
(415) 656-2890

Finally Technologies
25 Van Ness Ave.
San Francisco, CA 94102
(415) 564-5903
FAX (415) 626-4455

Golden Hawk Technology
427-3 Amherst St.
Nashua, NH 03063
(603) 424-0269

Great Valley Products Inc.
P.O. Box 391
Malvern, PA 19355
(800) 426-8957

HHT Electronics
2477 N. 450 W.
Harrisville, UT 84404
(801) 782-1841

Hypertek/Silicon Springs
120-1140 Austin Ave.
Coquitlam, British Columbia
CANADA V3K 3P5
(604) 939-8235

Impulse, Inc.
6870 Shingle Creek Parkway #112
Minneapolis, MN 55430
(612) 566-0221

Inner Connection Inc.
12310 Brandywine Road
Brandywine, MD 20613
(301) 372-8071

Interactive Microsystems
P.O. Box 1446
Haverhill, MA 01831
(617) 372-0400

Kent Engineering & Design
P.O. Box 178
Mottville, NY 13119
(315) 685-8237

Kline-Tronics
10 Carlisle Ct.
York, PA 17404
(717) 764-4205

Kurta
3007, Chambers East
Phoenix, AZ
(602) 276-5533

Lightning Publishing
1821 N. Ohio St.
Arlington, VA 22205
(703) 534-8030

Logical Design Works, Inc.
780 Montague Expwy., #403
San Jose, CA 95131
(408) 435-1445

Megatronics, Inc.
P.O. Box 3669
Logan, UT 84321
(800) 232-6342

Michigan Software
43345 Grand River
Novi, MI 48050
(313) 348-4477

MicroBotics, Inc.
811 Alpha Drive, Suite 335
Richardson, TX 75081
(214) 437-5330

MicroIllusions
17408 Chatsworth St.
Granada Hills, CA 91344
(818) 360-3715

Micron Technology, Inc.
2805 E. Columbia Rd.
Boise, ID 83706
(800) MICRON-1
(208) 386-3800

Microsearch Inc.
9896 Southwest Freeway
Houston, TX 77074
(713) 988-2818

Mimetics Corporation
P.O. Box 1560
Cupertino, CA 95014
(408) 741-0117

New Tek
115 W. Crane St.
Topeka, KS 66603
(913) 354-1146

Pacific Peripherals
P.O. Box 14575
Fremont, CA 94539
(415) 651-1905

Phoenix Electronics, Inc.
P.O. Box 156
Clay Center, KS 67432
(913) 632-2159

Pioneer Computing
2469 E. 7000 South #200
Salt Lake City, UT 84121
(801) 942-1174

Progressive Peripherals & Software
464 Kalamath St.
Denver, CO 80204
(303) 825-4144
FAX (303) 893-6938

R & D L Productions
11-24 46th Avenue, 2A
Long Island City, NY 11101
(718) 392-4090

Second Source Systems Inc.
501 Business Parkway
Richardson, TX 75081
(214) 680-8394

Spirit Technology Corp.
220 W. 2950 S.
Salt Lake City, UT 84115
(800) 433-7572
(801) 458-4233

Studio 7
P.O. Box 4727
Overland Park, KS 66204
(913) 649-0491

Sunrize Industries
3801 Old College Road
Bryan, TX 77801
(409) 846-1311

Supra Corporation
1133 Commercial Way
Albany, OR 97321
(503) 967-9075

Visual Aural Animation
P.O. Box 4898
Arcata, CA 95521
(707) 822-4800

Wollner Associates
3306 Horseman Lane
Falls Church, VA 22042
(703) 533-1236

Zebra Systems Inc.
78-06 Jamaica Ave.
Woodhaven, NY 11421
(718) 296-2385

•AC•

Proletariat Programming

A Look at Freely Distributable Compilers for the Amiga

by Patrick Quaid

The Amiga community is blessed with several freely redistributable compilers of various languages. Reviews and articles on these compilers almost always include the caveat that, while the compiler in question is useful to help beginning programmers learn the language, a commercial compiler is a minimum requirement for writing real programs.

But what are real programs? Perhaps they are programs someone else might need, programs that do some work, provide a service, or even play a game. Can something like this be written using a freely distributable compiler? Can those who would rather not spend hundreds of dollars on a language system participate in Amiga programming? You may be surprised.

Let us take a look at three of the freely distributable compilers available to Amiga programmers. (There are many others, but this article is not a book—at least, not *quite.*)

ETHZ Modula-2

ETHZ Modula-2 is the Swiss Federal Institute of Technology's Modula-2 compiler. It is included on Fred Fish disk #24 and was reviewed in *Amazing Computing* V1.10. I hope to cover different ground than that article did. The compiler is in the same vein as the Pascal-s and Pascal-p compilers—a public domain program written in part by the designer of the language, Dr. Niklaus Wirth. Unlike the Pascal-s compiler, however, ETHZ Modula-2 produces true 68000 native code, although it does not create executable files. It is evidently related to a Macintosh Modula-2 compiler as well as the M2Amiga compiler from AMSoft.

Modula-2 contains virtually every control structure and variable type commonly available in popular languages, and ETHZ Modula-2 supports every aspect of Modula-2 that I was able to test. Sets are supported (limited to the standard 16 components), as are procedure variables, real numbers, and all the loop structures of the language.

Some standard library routines are provided, but many are not. The standard module InOut is provided, but only for string and character types. Neither WriteInt, WriteCard, nor the RealInOut module are included (the review in AC V1.10 provides a WriteInt procedure). MathLib0 is also absent. Modula-2, of course, is designed to make an expansion of functionality as easy as possible, so these functions can all be added by the programmer.

Contrasting this lack of standard libraries is a veritable wealth of interesting procedures and example programs. The most important is LibCall(), provided in the module AMIGABase. This procedure, coupled with a type Regs that mimics the 680x0 register set, allows programmers to access any system routine for which they know the offset, much like assembly programs. Other interesting procedures include Translate() and Narrate(), which are as simple to use as they appear, and dec68k(), which takes a procedure as an argument and basically acts as a disassembler. Examples are provided for each of these.

The example programs include working demonstrations of an Intuition menu, simple sprites, opening a CLI, using the requester, creating speech, sensing CTRL-C, and several uses of AmigaDOS, including a type of Info command and a date printer. Also included are routines for (believe it or not) using the printf, sprintf, and fprintf routines in the Exec library from Modula-2. These examples are provided in lieu of documentation, but working examples are a programmer's best friend. Along with this lack of documentation, there's no explanation for the error numbers generated by the compiler. For what it's worth, I have included my interpretations of the errors I found.

The system obviously lacks a linker. The compiler produces object modules which are loaded and run by a bulky program called ALoad. Even the compiler is run through ALoad, which provides functions that would otherwise be undertaken by a combination of a linker, loader, and startup code. It loads all required modules, links them in memory, opens a window for standard input and output, etc. In some ways, this loader is novel. For instance, each object module is loaded only once, much like the Resident command. If another program requests the same module, it is linked in memory rather than loaded from the disk.

(continued)

This scheme raises some obvious questions about memory and cluttered disks, as well as some more subtle concerns. Since a compiled program does not keep a unique copy of an imported module, it is vulnerable to changes in that module. Obviously, Modula-2 must keep track of which version of a module is referred to by other modules. For example, recompiling the definition file of the module Clock invalidates all other Clock files, even if no changes were made. Thus the compiler, which uses the Clock module, will not be able to load properly. To fix this, the Clock implementation module and the compiler would have to be recompiled. However, the source is not provided for either of these, so be sure to keep copies of all the object modules.

ALoad is more annoying than necessary because it is case sensitive on its input line. It will terminate if a file is not properly capitalized or spelled, or not found, releasing all those painstakingly collected modules. It also requires that the module name exclude its .OBM extension. After it loads its object modules with ALoad, the compiler presents the user with a prompt. It is not case sensitive for its input line, but it does require that any extensions be included. Since it is difficult to keep these inconsistent formats straight, it does not terminate if a file is not found. Just to be irritating, the compiler accepts virtually anything but a letter, number or period as an end-of-line marker for its input line. The Return key is included in this set, as it should be, but so is most punctuation, all the control combinations, and the backspace key, so watch your fingers. Regardless of

ETHZ Modula-2 Error Numbers

My thoughts as to the meanings of the various ETHZ Modula-2 error numbers follow. Many of these are simply the type of mistake that caused the problem, and particularly questionable interpretations are marked by a question mark. Note that most of these were found by introducing deliberate errors. The others? Don't ask.

10	Identifier expected.
11	Comma expected.
12	Semicolon expected.
13	No TYPE given for parameters in PROCEDURE declaration.
14	Period expected.
15	Missing Parenthesis.
18	= expected.
19	:= expected.
20	Misplaced PROCEDURE declaration.
24?	Lower-cased reserved word.
27	THEN expected.
29	CARDINAL value greater than maximum.
31?	Unexpected keyword or symbol.
32	No dimensions given in ARRAY declaration.
34	Unfinished statement.
35	Missing preceding statement, like *) without (*
36?	Preceding keyword required (ie RETURN outside of PROCEDURE)
38	Unexpected identifier.
39	Parameter not expected, or EXIT in main module.
41	REAL value overflow (no underflow exists).
42	Block not completed.
44	Constant expected.
45?	Semicolon expected.
50	Undefined identifier.
52	Undefined TYPE.
57?	RECORD expected.
63	ARRAY dimension too large.
64	Too many parameters in PROCEDURE call.
65	Missing parameters in PROCEDURE call.
77	Names do not match at end of block.
81	Incompatible TYPEs in conversion, ie REAL(IntVar).
85	Modified IMPORT file exists.
88	IMPORT module does not exist.
100	re-TYPEd VAR.
108	Subscript out of range (a constant expression).
109	Bracket used instead of parenthesis.
110	Undefined field of a RECORD.
112	Integer less than -32767.
114?	SET TYPE expected.
115?	SET TYPE expected.
117	TYPE mismatch.
118	Boolean expression used in formula.
120	INTEGER operator used on REAL values.
132	Negative CARDINAL.
133	Assignments across incompatible TYPEs.
135	BOOLEAN expected.
136	Missing operator.
137	Incorrect TYPE in PROCEDURE call.
144	Incorrect TYPE in RETURN statement.
145	TYPE expected.
200	REAL operator used on INTEGER values.
208	INTEGER overflow (ie IntVar:= MAX(CARDINAL)).
209	SET of more than 16 elements.
210	ARRAY too large (in total size?).

the success of the compilation, the compiler returns its input prompt, a handy feature all compilers should incorporate.

With a capable linker, this compiler might be my favorite. It is fairly fast and provides the tools needed to write complete, rock-solid interfaces to all Amiga routines; all in the context of Modula-2, which I find the easiest language to develop with. The compiler handles all of Modula-2 with no apparent idiosyncrasies. But imagine releasing a program written with this system. The end user would need to create something called M2:, then copy all the required object modules into it. The user would then need to run ALoad and type in the name of the program with proper capitalization, at which time ALoad would create a full sized window on the Workbench screen, which could be a waste of memory. There is no reasonable way to run an ETHZ Modula-2 program from the Workbench. Even with a batch file to do the bulk of the work, it would have to be one terrific program to justify the trouble.

If you are writing an error lister or similar program for use within the ETHZ Modula-2 environment, none of this presents much of an obstacle. In fact, if these programs were written in ETHZ Modula-2, most of the need for a second CLI would be removed. Perhaps someone can use the system to write even the most modest of linkers. Until then, it is certainly possible to write large scale programs with ETHZ Modula-2; it's just not possible yet to let anyone else use them.

PDC

PDC is a freely distributable optimizing C compiler, originally written by Matthew Brandt for a UNIX system and moved to the Amiga environment by Jeff Lydiatt. It is available on Fred Fish disk #110, which also contains an assembler and Blink version 6.7. PDC produces 68000 assembly source code rather than object code directly, which is why an assembler is included.

The introductory file on the Fish disk notes several bugs and deficiencies of the compiler. In fact, the author claims in the very first paragraph that it will not work—a claim I am happy to contradict.

Actually, PDC has several commendable features. It is small (just over 50K on disk. PDC, the a68k assembler, and Blink all fit on the RAM: disk of a 512K machine comfortably). It is fast, and it produces very compact and fast code. It is also quirky, buggy, and comes with no header files, little in the way of function libraries, and nothing playing the role of Amiga.lib. Note that it is *free.*

PDC accepts most C contructs, but has notable limitations. All constants are considered long words: meaning 4-byte values. (I'm afraid there will be quite a bit of assembly talk in this), which can be a problem when passing parameters. Apparently, Global variables are not allowed. The compiler simply will not accept the concept of a void function (it will not accept the word void, but omitting it does the trick).

There is no shortage of similar quirks, but of course they are not fatal. But some are. Several times, through seriously deranged syntax errors I brought on the Guru. Nevertheless, these can be worked around, because the PDC programming environment (as it were) is remarkably flexible.

In a second-hand way, PDC subscribes to the standard Amiga object files and libraries. This means, for instance, that the programmer need not write his or her own printf() or puts() functions; they can be found in Amiga.lib. This also means that a standard linker like Blink can connect the C code produced with PDC to properly written assembly (or possibly Lattice C) object code you may have lying around. The point is that nearly any deficiency can be fixed, with varying numbers of backflips. One quick example would be this assembly code:

```
                XDEF      _VariableName
                SECTION   BSS
_VariableName   ds.l      1
                END
```

Assembled and linked to your program, this creates what amounts to a global variable called VariableName to a C program that includes a line like "extern int VariableName" where appropriate. Note that similar extern statements are not required for external functions.

As noted above, PDC is an optimizing compiler, which is no idle boast. Oft-used variables and procedure addresses are automatically housed in registers, and several arithmetic operations are condensed quite effectively. For example, PDC will convert a remainder operation into a 68000 AND instruction if the second operand is a power of two. Thus Var=Var % PowerOf2 becomes:

```
        MOVE.L   Var,d0
        AND.L    #PowerOfTwo-1,d0
        MOVE.L   d0,Var
```

This is up to seven times faster than the normal algorithmic method involving a DIVS instruction. Oddly, this optimization is carried out only on an equation in the above form; something like Var%= PowerOf2 is always figured algorithmically. The sequence is unfortunately never converted to the single instruction "AND.L #PowerOf2-1,Var", although additions, subtractions and assignments can take that form.

Similar optimizations are wrought upon multiplication and division operations, using shifts in place of

(continued)

MULS and DIVS when possible. This results in a similar savings in time, but the same restriction applies. The moral of this story is always to use the form "Var= Var * Constant" when the constant is a power of two. Also note that in PDC this form is never slower than the "Var *= constant" form. PDC also combines constants when possible, making, for example, "4*4" in the source simply "16" in the assembly produced.

Having raved about the optimization, I direct your attention to the Fibonacci benchmark results, in which the optimized version is actually slower. The only optimization wrought upon this program is storing variables in registers, which in this case saves very little time. Most of the work in this benchmark is making calls, during which register variables must be pushed onto the stack, whereas normal variables are already on the stack. The negligible savings in computation time is more than offset by the added time of saving registers. Not a very likely situation, but something to look out for.

Like most modern languages, all variables not declared static are held on the stack. This includes registers, which, as I mentioned, are saved when necessary. Since PDC assigns its own names—which it does not export—to static variables, and since it does not allow its own global variables, variable names are never exported. Those of you who have used Amiga.lib before know that it requires someone to export something called IntuitionBase in order to access Intuition functions (or GfxBase for Graphics functions, etc.). There are several ways to get around this. I decided to rewrite Astartup.asm so it opens Intuition and Graphics in the openDOS procedure, then closes them in the exit code. Astartup.asm is included on the disk with PDC, but you would play with whatever procedure you use. If you are not as lazy as I am, you could use the sort of global variable kludge listed above, which could be more efficient at run time.

One flagrant bug I found is in procedure calls. If the first argument of a call (the last one pushed on the stack) is an expression, as in foo(num + 1), the PDC generates incorrect code, which takes the form:

```
MOVE    d0,-(a7)
JSR     _foo
```

d0 here holds the result of the expression, and could be a different register. The first problem is that there is no length indicator on the MOVE instruction: a68k in this case assumes a word, although an int or long parameter should be a long word. The second problem is that the last parameter is not removed from the stack. There should be a statement like "ADDQ.L #4,a7" directly after the call. If there was more than one parameter, a statement to this effect is already present, but must be increased by four. The obvious fixes: either store the expression in a variable and use that as an argument, or edit the assembly code produced. Note that since this affects only the first argument, normal printf() calls always work properly.

I also cannot seem to get the switch() function to work properly. Blink complains that something called _c%switch has not been supplied by anyone. The documentation does not mention this so it is probably my mistake, but I have included a short assembly program called cswitch.asm that will fix the problem if linked to the code that uses a switch/case statement. This illustrates not only how to fix bugs of this sort, but also how to add more useful assembly functions to your PDC programs.

cswitch.asm

Examine the code immediately before and after the call to _c%switch in the assembly file (produced by PDC) in order to get a better idea of how this works. Note that most subroutines called from PDC should set up a local stack frame and save all registers used; cswitch.asm does neither.

```
        XDEF     _c%switch
        CODE
_c%switch
        move.l   (a7)+,a0 ;POP the table addr
1$
        tst.l    (a0)                ;default case?
        beq.s    2$                  ;if so, go away

        cmp.l    4(a0),d0 ;d0 holds compare val
                 beq.s    3$         ;if equal, return
                 addq.l   #8,a0      ;go to next entry
                 bra.s    1$         ;and try again
2$
                 addq.l   #4,a0      ;for default, shift
3$
        move.l   (a0),a0  ;get return address
        jmp      (a0)
```

To write complete programs with PDC, you will need to write header files (which can be culled from the ROM Kernel Manuals as the need arises). Then get a library (like Amiga.lib if you own Lattice C or the Metacomco Assembler, or smalllib from Fred Fish disk #92). You can use the assembler from the Fish disk #110 or any other Motorola standard assembler. Later, you will want some standard functions like gets(), which you can write yourself as you need them.

When PDC tells you something is amiss even though you know it's correct, don't fight it. Try to reshape it or attack it from another direction. When you've pinned down the problem, write it down somewhere and try to fix it or just avoid it. You will need plenty of patience, and no doubt some assembler expertise. Remember that PDC, like many compilers, is not terribly adept at recovering from errors; the first error normally triggers many false ones.

The fruit of this labor will be very small code, which, for many operations, is about as fast as you'll find, since it does no error checking. If you don't mind meddling with the assembly code you can easily make it faster than any high-level language. Your source code might have to be massaged through the compiler and will not be able to use the full power of C (like floating point numbers), but you should be able to produce fairly complex programs. The only bugs I have run across that aren't in the documentation are listed above. And all the problems I have found had fairly simple fixes. The tallest hurdle in using PDC is convincing the compiler to recognize constructs, but anything missing can be added in assembler.

If you like programming in assembly, this compiler is especially useful. You can write all your control structures in C, then insert your assembly code where necessary. Likewise, you can write tricky equations in C to get a model for an assembly procedure. PDC supplies no comments in its assembly output, but the code is easy to digest and even learn from. PDC removes much of the tedium involved and, if you are like me, cuts down significantly on logic errors. In his documentation for Draco (see below) Chris Gray writes that anyone who really tinkers with compiler-produced assembly code has to be crazy. Well, I do like to tinker with it, so I am in a position to agree with him wholeheartedly.

If C is your job or your life, you'll need a commercial compiler so you can complain with conviction. Those of you interested in writing compilers could undoubtedly learn a thing or two from the C source code (Manx flavored) included on the Fish disk. It would be not only an interesting exercise, but also a great benefit to the Amiga community if some C wizard would undertake the next revision of PDC.

But what about those of us who simply want to write C programs? PDC is no competition for commercial offerings, but it does most of what it is supposed to do. My background is in the non-toxic languages, but with PDC, I feel confident that I can write full scale programs—programs that are smaller and faster (and in some ways nastier) than those of any other language I know. My intuition.h file is expanding, my graphics.h file is nearly complete, and my other header files are growing as necessary, so I'm well on my way to a complete C system; no apologies, no excuses.

Draco

In 621 B.C. Draco (also known as Dracon) began passing laws in Athens that called for the death penalty for even minor crimes. In memory of him, the term Draconian is used describe a rigid law or cruel punishment. In an age when programming languages like Pascal and Ada are named after people, Chris Gray apparently named his pet language after Draco. Of course, the same word in Greek means serpent, so that's only a guess.

Draco is a structured programming language written by Gray for CP/M systems several years ago, and ported to the Amiga just recently. The compiler, libraries, header files, documentation and example programs are available on Fred Fish disks 76 and 77.

Draco is similar to the rest of the common procedural languages like Pascal, C, Modula-2, and AmigaBASIC. At first glance it appears most like C, with liberal doses of Pascal. The common control structures such as "if" statements (with "else" and "else-if" clauses), "case" statements, "for" loops and "while" loops (which are flexible enough to act as true "repeat" loops) are included, as one would expect. Standard variable types like long and short integers, characters, Boolean values, pointers, arrays, and structures are supported, as are less common types like enumerated types and unions. Draco supports normal and long real numbers and an operator type, but these are not yet supported in the Amiga version.

Indeed, Draco has much more in common with the more popular languages than it has differences. Most of its features can be found in some other language, although none of the other languages has a similar combination. For instance, simple output is handled with write() and writeln() calls, just like Pascal, although in Draco the field widths can be specified with variables. These, along with the related read() and readln() calls, make up the simplest terminal input/output this side of BASIC. Declaring variables and constants is much like C, although designed more logically than C, in my opinion. Procedure declarations are

(continued)

similar to Modula-2, with the parameter types listed within the parentheses and function types listed at the end of the line. Equations look most like C, with more symbols than are absolutely necessary, although at least "and" and "or" can be in English. Draco, like Pascal and Modula-2, makes sure variable types in assignments and equations are the same, which might be where Draco got its name. (Some C programmers consider loose type checking a Constitutional right.)

Any language that checks types strictly must have a method for circumventing that control. Draco uses the function "pretend," which takes an expression and a type as arguments, then pretends that the expression result is of the given type. Modula-2 has a more elegant solution, but "pretend" seems to do more conversion work. Let's face it, it is named realistically.

Draco also offers "if" and "case" expressions. The idea here is that a variable be assigned, for instance, to an "if" expression, which would look something like:

```
            Var:=  if OtherVar < 50 then
writeln("The other variable is ",OtherVar);
            OtherVar + 50
                      else
            OtherVar
                      fi;
```

Note the final line of each clause is simply an expression. The meaning of this code is clearer, I think, if "Var:=" is simply inserted before each of these expressions—the statement form. The expression form may have a convenient use that I have yet to run across. Function values are returned similarly, with a line consisting only of an expression.

What's Draco???

What follows is a simple Draco program that opens a window on the Workbench screen, then waits for the close message from Intuition.

```
 #include:exec/miscellaneous.g
#include:exec/ports.g
 #include:intuition/miscellaneous.g
 #include:intuition/window.g
 #include:intuition/screen.g
 #include:intuition/intuimessage.g

proc main()void:

*Window_t        window;
 *NewWindow_t    newwindow;
*Message_t       quit_message;

 if OpenExecLibrary(33) ~= nil then
      if OpenIntuitionLibrary(33) ~= nil then
            newwindow:= &NewWindow_t( 160, 50, 320, 100,
                            FREEPEN, FREEPEN, CLOSEWINDOW,
                             SMART_REFRESH|ACTIVATE|WINDOWSIZING|
                             WINDOWDEPTH|WINDOWCLOSE|WINDOWDRAG|
                            NOCAREREFRESH, nil, nil, nil, nil, nil,
                            50, 100, 640, 200, WBENCHSCREEN );
            newwindow*.nw_Title:= "Simple Window";
            window:= OpenWindow(newwindow);
          if window ~= nil then
                quit_message:= WaitPort(window*.w_UserPort);
                quit_message:= GetMsg(window*.w_UserPort);
              CloseWindow(window);
         else
              writeln("Couldn't open the window");
         fi;
           CloseIntuitionLibrary();
     fi;
      CloseExecLibrary();
fi;
corp;
```

Draco's conditional compilation facilities are similar to those of Modula-2, in which an "if" statement that evaluates to "false" (either as a constant or constant expression like sizeof()), during compilation is simply not compiled at all. This allows the inclusion of a Boolean constant called, for instance, DEBUG, which can be set to "true" in development. Debugging code can then be included in the source surrounded by an "if DEBUG" type statement. When DEBUG is set to "false", the debugging code is not included and does not inflate the size of the resulting program. Draco also includes a method for issuing actual compiler errors from the source code, so the bases should be well covered.

A programmer who has used Pascal, Modula-2, or even AmigaBASIC, and who has even the most rudimentary knowledge of C can learn Draco in a day. It's not *easier* than most languages, but it's not more difficult, either. It amounts to using the same old tools and methods with a variation of the old syntax rules. A simple Draco program that opens a window and waits for the user to close it is included to provide a limited illustration of Draco.

On the two distribution disks are a complete set of Draco include files, arranged in files and directories similar to the assembler and C sets, and written according to Gray's proposed Draco conventions. There are also many libraries to be linked to Draco object files. These all have corresponding include files, and mostly concern Amiga functions like the dos.lib or intuition.lib. Others provide extra Draco functions. Crt.lib, for example, provides a long list of screen handling functions, from simple things like placing and finding the cursor to a full set of form definition and input/output routines. Along the way are simple functions for asking questions, centering and highlighting text, and even turning automatic handling of the Control-C combination on and off.

These and other functions provided with the distribution show the work Gray has put into the system over the years. There are, for example, a set of string handling functions, many input/output options including simple random file access, memory management tools, and time functions. The depth and breadth of the support is stunning. The balance of the distribution is documentation, helpful batch files, an editor written in Draco and several example programs, an absolute requirement for an unfamiliar language. For one whopper of an example program, see Empire on Fred Fish disk #118; about 350K of source code is included.

Draco is designed to be linked by a smart linker that includes only those routines actually called. Gray has yet to port the Draco linker to the Amiga, however, so the system currently uses Blink. Unfortunately this means that Amiga Draco programs are very large, since some of the lavish support provided by Draco might go unused by many programs. Gray does write in his introduction that he plans to port his linker, which should help out significantly. Gray also notes that "code quality is not so good" with this version, which is borne out by the benchmarks. This implies that he plans to speed it up somewhat, which is always welcome.

The compiler itself is apparently written in Draco, although the source is not provided. This has a pleasant effect: as the Draco environment improves, the compiler itself gets smaller and faster. The compiler is already less than 80K on the disk, and can join Blink on a 512K Amiga's RAM: disk comfortably. As the benchmarks show, the compiler is quite fast. Contrary to what its name might imply, Draco flags errors with English sentences pulled from a text file which can be edited as necessary. Those errors are, in turn, fully explained one by one in a documentation file.

The Amiga version of Draco, it appears, is a shadow of its more mature CP/M sibling, although of course it is complete as it stands. The Draco linker provides a form of modularity, complete with automatic startup and termination procedures (going Modula-2 one better). The documentation mentions the existence of a companion assembler, disassembler, and librarian for the Draco system. When and if these tools are ported to the Amiga, Gray's Draco should be among the most powerful Amiga programming environments available. As it stands, it is a full featured compiler whose only real drawback is the unfamiliarity of the language.

Having given the Amiga community a taste of what it can expect from Draco, perhaps Gray could release the complete version as a more commercial product. Or, in a fit of marketing brilliance, he could release just the additional components, with the addition of a debugger, as commercial products (or would that be too confusing?). In any case, one would hope that the rest of the system finds its way into the Amiga. At least my Amiga.

Benchmarks

The benchmarks are not, of course, meant to compare the three compilers directly; there are many more important factors to consider. In fact, no particular effort was made to create equivalent categories for the compilers. Instead, compare the results of the benchmarks to what you require in a compiler. Run time results for the TDI Modula-2 compiler, with all error checking disabled, are provided in order to present a broader basis for comparison. No other figures are provided for the TDI compiler, since they would have no relation to apparently analogous results for the free compilers.

The Math benchmark just carries out simple math operations a ridiculous number of times. The multiplication, division, and remainder operands were specifically chosen so that PDC would not optimize them out of existence. Be assured that if powers of two were used, the PDC optimized run time would be much faster than it is.

The Fib benchmark calculated the 30th Fibonacci number by the slow and inefficient recursive method, to test the efficiency of function calls for each compiler. Believe me, this program makes a lot of calls.

The Sieve and Prime benchmarks were pulled from Amazing Computing's review of TDI Modula-2 in V1.9. The TDI results are also from that issue. These benchmarks are included because the most common and important use of computers is finding prime numbers.

The Prime benchmark normally prints hundreds of prime numbers to the screen, which makes the benchmark's final time rely heavily on printing time. ETHZ Modula-2 had to resort to the WriteInt() procedure from *Amazing Computing* V1.10 for its output, and since its slowness definitely skewed the results, I ran another version of the program that computes the primes but does not print them. This has the effect of stealing the last remnant of utility from the benchmark itself. It also shows something about the relative printing speeds.

(continued)

In each case, the run times for ETHZ Modula-2 were cursor-to-cursor times for the first run of the program with only the compiler's modules already in memory. Subsequent runs would be quicker since no loading would be necessary.

The build time for PDC and Draco are the cursor-to-cursor run times for a batch file that compiles and links a program using the RAM: disk as much as is convenient. For PDC this includes an assembly step, of course.

Conclusions

Of these compilers, Draco is the best suited for general purpose programming on the Amiga. The unique language, regardless of its relative merits, must be considered a negative factor. What makes this the superior environment is the inclusion of complete headers and libraries, as well as fertile support functions. The compiler itself is solid, with no obvious bugs and only real numbers and operator types as unimplemented features. The code is slow and fat, and will hopefully speed up and slim down, but more importantly, it works as expected. As the CP/M system moves to the Amiga, Draco will become stronger and more complete.

As it turns out, PDC is the only real challenger to Draco for general purpose programming. It has the advantage of being the most popular language among Amiga programmers, and also produces faster and smaller code then almost anything else. There are no header files or libraries included, however, and the limitations of the compiler make most of the existing header files inconvenient. Although its problems can normally be overcome, the compiler itself is somewhat buggy and fails to recognize several constructs. Unfortunately, most C programs need to be changed considerably for PDC to compile them. If you need or want C, however, this is a useful program that becomes much easier to use effectively as you get more experience with it.

ETHZ Modula-2 simply cannot be used for general programming, since every program requires ALoad and something called M2: (perhaps this would work better on a hard disk). This is unfortunate since it is a very complete version of Modula-2 that includes a handy ability to access all the Amiga's features. Modula-2 is, in my opinion, the easiest language yet to write large programs with, and this implementation does nothing to change that. For those of you who want Modula-2, a commercial compiler is the only real game in town. Happily there is a good selection for Amiga owners.

So choose your poison. Writing complete, useful programs need not be the exclusive domain of the aristocrats who can afford a compiler costing half as much as their computer did. There are free compilers out there just waiting to be exploited. In addition to these three, there is a related C compiler, a number of assemblers, a Forth implementation, XLISP, several specific purpose languages like the Adventure Definition Language, and others. Use them, abuse them, but if they help you, make sure you take care of their authors.

•AC•

Benchmark Results

		Math	Fib	Sieve	Prime
ETHZ Modula-2					
	Load Compiler	30	30	30	30
	Compile 14	10	13	9	
	Run 282	101	10	70	
	Run w/o printing		34		
PDC					
	Compile & Assemble				
	from DF0:	22	23	31	24
	from RAM:	10	10	11	10
	Build in RAM:	39	35	42	41
	Run				
	Optimized	161	95	5	39
	Not Optimized	285	82	11	49
	Run w/o printing		30		
	Size	1460	1948	10212	1896
Draco					
	Compile from RAM:	5	6	7	6
	Build in RAM:	39	41	43	42
	Run	356	136	9	46
	Run w/o printing		42		
	Size	10468	18124	16344	16100
TDI Modula-2					
	Run	309	72	7	54

the BIG Picture

by *Warren Ring*

The Unified Field Theory Continued!

Two months ago, we started a series of articles on interfacing among a number of assembler functions (a collection known as a toolbox) with unified fields. This article is the last in a three-part series describing these functions. I've included source code covering disk I/O, console I/O, scanning a line of text for individual words, string manipulation, Integer/ASCII conversion, and text display.

This month we'll finish this series with source code and examples. Add the text shown in this month's MACROS.ASM and WARLIB.ASM files to the text shown in last month's column. Later in this article, I'll discuss how you can obtain a copy by modem.

For those of you just joining us this month, the toolbox uses a data structure I call the "string buffer" for getting data to and from the routines in this toolbox. A string buffer can contain text strings, but it can just as easily be used for disk buffers. It may contain either binary or ASCII data, and is structured to have overrun protection and fast manipulation. It consists of a long integer indicating the size of a data area in bytes, followed by a long integer indicating the current usage of that data area, in bytes, followed by that data area. String buffers unify the interfaces to all functions in this toolbox, and simplify many coding operations assembler programmers find difficult.

The functions we'll cover this month can best be illustrated by the two example test programs.

Example One: Character Conversion

Listing 1 shows examples of character conversion from ASCII to hex ASCII. The program allows you to enter a phrase and then displays the individual ASCII characters and their hexadecimal equivalents for each character you've entered. It demonstrates how the toolbox converts nybbles (4 bits), bytes, words, and long words from integer format to 1, 2, 4, or 8 character format, respectively. These character strings can then be displayed on the console or combined with other text. Listing 2 shows a sample run of this program.

Example Two: Integer/ASCII Conversion

Listing 3 shows how you can convert a signed integer from an ASCII string to binary form and back again. The program also shows you how to display a binary integer as 8 hex ASCII digits. You can use this facility to get decimal signed integer responses from the console, convert the values to long integers, convert values from signed integers to decimal ASCII strings, and display these strings on the console. Listing 4 shows a sample run of this program. This operation can be tremendously useful in communicating values to and from the console.

Parting Words

This entry wraps up the first in what I hope will be a series of assembler toolboxes. I'm going to take a few months off to work on another project. Eventually, I would like to get into serial and parallel port usage, multi-tasking, and floating point support in the next toolbox. These would enable you to write your own BBS and resident print spooling routines. In subsequent toolboxes, I would like to cover graphics support.

If you missed the last two columns, and would rather get a copy of these files electronically, the files are currently available under the name ASMTOOL1.ARC on People Link and BIX, and on the two BBS systems I regularly visit: the Los Angeles Amiga User's Group BBS (LAAUG Line) at (213) 559-7367, and the "1939" BBS at (818) 368-4248. If you have questions or comments, you may contact me at either of these boards or through this magazine. Good luck!

Listing 1: TEST1.ASM

```
; TEST1.ASM by Warren A. Ring
;
; This program shows examples of character
; conversion from ASCII to hex ASCII.  It allows
; you to enter a phrase, then it displays the
; individual ASCII characters and their
; hexadecimal equivalents for each character you
; entered.
```

(continued)

Magnetic Images™

CLIP ART !

For *AMIGA*™
Over 100 high resolution
images on each disk.

Disk 1 : Computer, Office, Music, School, Travel, Trans.
Disk 2 : Business, Sports, Animals, Party, Religious
Disk 3 : Food, Borders, Medicine, Old West, Newsletter
Disk 4 : Hands, Seasons, Pirates, Tools, Personal, America
Disk 5 : Theater, Corners, Zoo, Menu, Outdoor
Disk 6 : Adman's Special: Computer Products

$19.95 per disk (Add $2.50 P & H per order)

Magnetic Images Co.
P.O. Box 17422, Phoenix, AZ 85011
(602) 265-7849
Dealer inquiries welcome.

```
   section code

   include "macros.asm"

   Start       ;Perform startup housekeeping
X1 Display <'Enter a phrase: '>
   ReadCon #Word    ;Get a line from the console
   StrLen  #Word    ;If no characters were entered,
   BEQ     X99      ; then jump to X99
   SetScan #Word    ;Set to scan the console line
   Display <'Characters are:',LF>
X2 Scanc   #Char    ;Scan the console line for the
                    ; next character
   StrLen  #Char    ;If no character is available,
   BEQ     X1       ; then jump to X1
   WritCon #Char    ;Display the character
   Space            ;Display a space
   ItoHA2  Char+8,#HexCode;Convert the character
                    ; from ASCII to hex ASCII
   WritCon #HexCode;Display the hex ASCII code
   Crlf             ;Display a CR/LF
   BRA     X2       ;Jump to X2

X99
   Exit    ;Perform ending housekeeping, and exit

   include "warlib.asm"

   section data

   StrBuf  Word,16
   StrBuf  Char,1
   StrBuf  HexCode,2

   end
```

Listing 2: TEST1 sample run

```
1>test1
Enter a phrase: hello, world
Characters are:
h 68
e 65
l 6C
l 6C
o 6F
, 2C
  20
w 77
o 6F
r 72
l 6C
d 64
Enter a phrase:
1>
```

Listing 3: TEST2.ASM source code

```
; TEST2.ASM by Warren A. Ring
;
; This program shows how you can convert an
; integer from ASCII string to integer form and
; back again.  It also shows you how to display a
; binary integer as 8 hex ASCII digits.

   section code

   include "macros.asm"

   Start            ;Perform startup housekeeping
X1 Display <'Enter a decimal number: '>
   ReadCon #Word    ;Get a line from the console
   StrLen  #Word    ;If no characters were entered,
   BEQ     X99      ; then jump to X99
   Display <'The hexadecimal equivalent is: '>
   AtoI    #Word,Value ;Convert the string from
                    ; ASCII to an integer
   ItoHA8  Value,#HexCode;Convert the integer to
                    ; 8-character hex ASCII
   WritCon #HexCode;Display the hex ASCII string
   Crlf             ;Display a CR/LF
   Display <'Enter a hexadecimal number: '>
   ReadCon #Word    ;Get a line from the console
   StrLen  #Word    ;If no characters were entered,
   BEQ     X99      ; then jump to X99
   Display <'The decimal equivalent is: '>
   HAtoI   #Word,Value ;Convert the string from
                    ; hex ASCII to integer
   ItoA    Value,#Word ;Convert the integer to an
                    ; ASCII string
   WritCon #Word    ;Display the ASCII string
   Crlf             ;Display a CR/LF
   BRA     X1       ;Jump to X1

X99
   Exit    ;Perform ending housekeeping, and exit

   include "warlib.asm"

   section data

   StrBuf  Word,16
   StrBuf  HexCode,8

Value  DS.L    1

   end
```

Listing 4: TEST2 sample run

```
1>test2
Enter a decimal number: 1234
The hexadecimal equivalent is: 000004D2
Enter a hexadecimal number: 50
The decimal equivalent is: 80
Enter a decimal number: -1
The hexadecimal equivalent is: FFFFFFFF
Enter a hexadecimal number: 7FFFFFFF
The decimal equivalent is: 2147483647
Enter a decimal number: -2
The hexadecimal equivalent is: FFFFFFFE
Enter a hexadecimal number: 80000000
The decimal equivalent is: -2147483648
Enter a decimal number:
1>
```

Listing 5: Macros.asm

(The following macro definitions should be appended to the list of macro definitions shown for this file in last month's column)

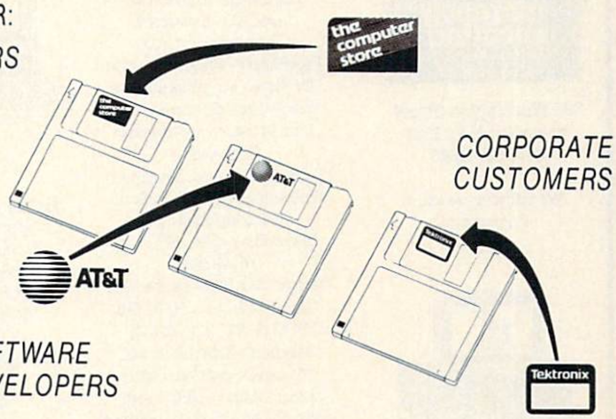
```
;SB = the location of a string buffer

Scanc      MACRO
           MOVEM.L A0,-(A7)
           MOVEA.L \1,A0
           JSR     Scanc_
           MOVEM.L (A7)+,A0
           ENDM

Scana      MACRO
           MOVEM.L A0,-(A7)
           MOVEA.L \1,A0
           JSR     Scana_
           MOVEM.L (A7)+,A0
           ENDM

HAtoI      MACRO
           MOVEM.L D0/A0,-(A7)
           MOVE.L  \1,A0
           JSR     HAtoI_
           MOVE.L  D0,\2
           MOVEM.L (A7)+,D0/A0
           ENDM

ItoHA8     MACRO
           MOVEM.L D0/A0,-(A7)
           MOVE.L  \1,D0
           MOVE.L  \2,A0
           JSR     ItoHA8_
           MOVEM.L (A7)+,D0/A0
           ENDM

ItoHA4     MACRO
           MOVEM.L D0/A0,-(A7)
           MOVE.W  \1,D0
           MOVE.L  \2,A0
           JSR     ItoHA4_
           MOVEM.L (A7)+,D0/A0
           ENDM

ItoHA2     MACRO
           MOVEM.L D0/A0,-(A7)
           MOVE.B  \1,D0
           MOVE.L  \2,A0
           JSR     ItoHA2_
           MOVEM.L (A7)+,D0/A0
           ENDM

ItoHA1     MACRO
           MOVEM.L D0/A0,-(A7)
           MOVE.B  \1,D0
           MOVE.L  \2,A0
           JSR     ItoHA1_
           MOVEM.L (A7)+,D0/A0
           ENDM
```

Listing 6: Warlib.asm

(The following library routines should be appended to the list of library routines shown for this file in last month's column)

```
Scana_

;This routine scans a string for the next
; alphanumeric word

;In:  ScanPointer => the next byte in the string
;                    to be scanned
;     ScanCounter = the number of bytes in the
;                   string remaining to be scanned
;     A0 => the dest buffer
;Out: ScanPointer, ScanCounter are updated
;
;Note: This routine skips over spaces, tabs, and
;      invisible characters including non-
;      alphanumeric characters

  MOVEM.L D0-D2/A0-A2,-(A7)     ;(Push registers)

  MOVE.L  (A0)+,D1;(Set D1 to the max length of
                  ; A$)
  MOVE.L  A0,A1   ;(Make A1 point to the current
                  ; length of A$)
  MOVE.L  #0,(A1) ;Set the current useage of A$
                  ; to 0
```

(continued)



"NEW" PRINTED 3.5" SHUTTERS

FOR:
VARS

CORPORATE CUSTOMERS

SOFTWARE DEVELOPERS

Permanent Identification of your Company, Information, or Products.

SOFTWARE DUPLICATION/PACKAGING

3.5" / 5.25" / HIGH DENSITY

WEST COAST TELECOM
CALL: (503) 620-1888

The Memory Location
Computers for the Home

396 Washington Street
Wellesley, MA 02181
(617) 237-6846

We sell what we know -
Commodore!

Amiga
Specialists!

New for the Amiga
PIXmate - great image conversion and processing!
Interchange - conversions
IntroCAD - a powerful 'introductory' CAD
PageFlipper - simple animation
TV Show - scripted animation
WordPerfect- the name says it!
Data Retrieve - new database from Abacus
Textcraft Plus - much improved
Professional Page - the ultimate in desktop publishing
FlickerFixer - 704 x 470 with NO flicker!
Micron 2Mb RAM for the 2000
Hard disk drives - 20-65 Mb, SCSI & ST, 3.5" & 5.25"
Monitors - Sony, Thomson, Magnavox and Commodore
Laser Printers - NEC's best!
Ports of Call * Galactic Invasion
Foot Man * Tele War
Softwood Write & File
Pro ASM * AssemPro
Graphic Studio
DIGIDroid
Hunt for Red October
Time Bandit * Hollywood Poker
GOMF!

New for the 64 / 128
Super Boulder Dash
Sesame Street titles
Sticky Bears series
geoProgrammer
Kung Fu I & II
Card Shark
geoPublish
Test Drive
SwiftTax
Sherlock
NeoFont
Ace 2
Ardok
The Train
Bazooka Bill
MISL Soccer
Rogue Trooper

Only What Works - Satisfaction Guaranteed!

Amiga / C-64 / C-128 - Authorized Repair Center
Hardware / Software / Printers / Peripherals / Accessories

```
   ADDQ.L  #4,A0   ;Advance the dest buffer
                   ; pointer (A0) to the first
                   ; data byte of A$
   MOVE.L  ScanPointer,A2;(Set A1 to the scan
                   ; pointer)
   MOVE.L  ScanCounter,D2;If the scan counter (D2)
                   ; is zero,
   BEQ     Scana_9 ; then jump to Scana_9
Scana_1
   MOVE.B  (A2)+,D0;Fetch the character at which
                   ; the scan pointer points,
                   ; and advance the scan pointer
   SUBQ.L  #1,D2   ;Decrement the scan counter
   CMPI.B  #$30,D0 ;If the character is alpha-
   BLT     Scana_3 ; numeric, then jump to Scana_2
   CMPI.B  #$3A,D0
   BLT     Scana_2
   CMPI.B  #$41,D0
   BLT     Scana_3
   CMPI.B  #$5B,D0
   BLT     Scana_2
   CMPI.B  #$61,D0
   BLT     Scana_3
   CMPI.B  #$7B,D0
   BLT     Scana_2

Scana_3
   TST.L   D2      ;If the scan counter is zero,
   BEQ     Scana_9 ; then jump to Scana_9
   BRA     Scana_1 ;Jump to Scana_1
Scana_2
   MOVE.B  D0,(A0)+;Place the character into A$,
                   ; and advance the dest buffer
                   ; pointer
   ADD.L   #1,(A1) ;Increment the current useage
                   ; of A$
   CMP.L   (A1),D1 ;If the current useage of A$ =
                   ; the max length of A$,
```

```
   BEQ     Scana_9 ; then jump to Scana_9
   TST.L   D2      ;If the scan counter is zero,
   BEQ     Scana_9 ; then jump to Scana_9
   MOVE.B  (A2)+,D0;Fetch the character at which
                   ; the scan pointer points,
                   ; and advance the scan pointer
   SUBQ.L  #1,D2   ;Decrement the scan counter
   CMPI.B  #$30,D0 ;If the character is alpha-
   BLT     Scana_4 ;numeric, then jump to Scana_2
   CMPI.B  #$3A,D0
   BLT     Scana_2
   CMPI.B  #$41,D0
   BLT     Scana_4
   CMPI.B  #$5B,D0
   BLT     Scana_2
   CMPI.B  #$61,D0
   BLT     Scana_4
   CMPI.B  #$7B,D0
   BLT     Scana_2
Scana_4

Scana_9
   MOVE.L  A2,ScanPointer
                   ;(Update the scan pointer)
   MOVE.L  D2,ScanCounter
                   ;(Update the scan counter)

   MOVEM.L (A7)+,D0-D2/A0-A2    ;(Pop registers)

   RTS             ;Return


Scanc_

;This routine scans a string for the next
; character

;In:  ScanPointer => the next byte in the string
;                    to be scanned
;     ScanCounter = the number of bytes in the
;                   string remaining to be scanned
;     A0 => the dest buffer
;Out: ScanPointer, ScanCounter are updated
;

   MOVEM.L D0-D2/A0-A2,-(A7)    ;(Push registers)

   MOVEQ.L #0,D0   ;Set the byte counter to 0
   ADDQ.L  #4,A0   ;(Make A0 point to the current
                   ; useage of A$)
   MOVEA.L ScanPointer,A1;(Set A1 to the scan
                   ; pointer)
   MOVE.L  ScanCounter,D1;(Set D1 to the scan
                   ; counter)
   BEQ     Scanc_9 ;If the scan counter is 0,
                   ; then jump to Scanc_9
   MOVE.B  (A1)+,D2;Fetch the character at which
                   ; the scan pointer points,
                   ; and advance the scan pointer
   SUBQ.L  #1,D1   ;Decrement the scan counter
   ADDQ.L  #4,A0   ;(Make A0 point to the first
                   ; data byte in A$)
   MOVE.B  D2,(A0) ;Place the character in the
                   ; buffer
   SUBQ.L  #4,A0   ;(Make A0 point to the current
                   ; useage of A$)
   MOVEQ.L #1,D0   ;Set the byte counter to 1
Scanc_9
   MOVE.L  D0,(A0) ;Set the current useage in A$
                   ; to the byte counter
   MOVE.L  A1,ScanPointer
                   ;(Update the scan pointer)
   MOVE.L  D1,ScanCounter
                   ;(Update the scan counter)

   MOVEM.L (A7)+,D0-D2/A0-A2    ;(Pop registers)

   RTS             ;Return
```

```
HAtoI_              ;BASIC Fn: I = HEX(A$)
                    ;          (D0)   (A0)

;This routine converts a hex-ASCII string to a
; binary integer
;
;Note: Conversion stops if a non-hex-ASCII
; character is encountered in the string

   MOVEM.L D1-D2,-(A7) ;(Push registers)

   MOVEQ.L #0,D2    ;Set the result (D2) to 0
   ADDQ.L  #4,A0    ;Set the limit value (D1) to
                    ; the current length of A$
   MOVE.L  (A0)+,D1;(Make A0 point to the current
                    ; useage of A$)
HAtoI_1
   MOVE.B  (A0)+,D0;If the first (next) character
                    ; in A$ is not a hex-ASCII
   JSR     IsHexASCII; character, then jump to
   BEQ     HAtoI_7 ;HAtoI_7
   CMPI.B  #'9'+1,D0;Convert the hex-ASCII
   BCS     HAtoI_2 ; character to a binary value
   ANDI.B  #$0F,D0
   ADD.L   #9,D0
   BRA     HAtoI_3
HAtoI_2
   SUB.L   #'0',D0
HAtoI_3
   LSL.L   #4,D2    ;Shift the result left 4 bits
   OR.L    D0,D2    ;'Or' the binary value into
                    ; the result
   SUBQ.L  #1,D1    ;Decrement the limit value
   BNE     HAtoI_1 ;If the limit value is not yet
                    ; zero, then jump to HAtoI_1
HAtoI_7

   MOVE.L  D2,D0    ;(Set D0 to the result)
   MOVEM.L (A7)+,D1-D2 ;(Pop registers)

   RTS              ;Return


ItoHA8_             ;BASIC Fn: A$ = STR8HEX$(I)
                    ;          (A0)         (D0)

   MOVEM.L D1,-(A7);(Push registers)

   MOVEQ.L #8,D1    ;Set the field width to 8
   JSR     ItoHA_   ;Convert the field

   MOVEM.L (A7)+,D1;(Pop registers)

   RTS              ;Return


ItoHA4_             ;BASIC Fn: A$ = STR4HEX$(I)
                    ;          (A0)         (D0)

   MOVEM.L D1,-(A7);(Push registers)

   MOVEQ.L #4,D1    ;Set the field width to 4
   JSR     ItoHA_   ;Convert the field

   MOVEM.L (A7)+,D1;(Pop registers)

   RTS              ;Return


ItoHA2_             ;BASIC Fn: A$ = STR2HEX$(I)
                    ;          (A0)         (D0)

   MOVEM.L D1,-(A7);(Push registers)

   MOVEQ.L #2,D1    ;Set the field width to 2
   JSR     ItoHA_   ;Convert the field

   MOVEM.L (A7)+,D1;(Pop registers)

   RTS              ;Return
```

INTERCHANGE ™

Share objects between Sculpt3D, VideoScape 3D & Forms in Flight

NOW YOU CAN...

- **Use Sculpt 3D to ray–trace VideoScape 3D objects**
- **Do Forms in Flight animations on Sculpt 3D objects**
- **Create VideoScape 3D objects using the Sculpt 3D interface**

Full Intuition interface for all InterChange functions

Object Disk #1 now available. Includes a Sculpt 3D font, plus lots more Sculpt 3D and VideoScape 3D objects!

InterChange master program plus Sculpt 3D and VideoScape 3D Conversion Modules, $49.95. Forms in Flight add–on Conversion Module, $19.95. InterChange Object Disk # 1, $19.95.

This product requires objects from Sculpt 3D and/or VideoScape 3D and/or Forms in Flight. It is not a standalone animation program.

To order, send check or money order. Please include $3.00 postage & handling. MA and WI residents add 5% sales tax. InterChange is a trademark of Syndesis. Sculpt 3D, VideoScape 3D and Forms in Flight are trademarks of Byte by Byte Corporation, Aegis Development and Micro Magic respectively.

SYNDESIS
20 WEST STREET
WILMINGTON, MASSACHUSETTS 01887
617–657–5585

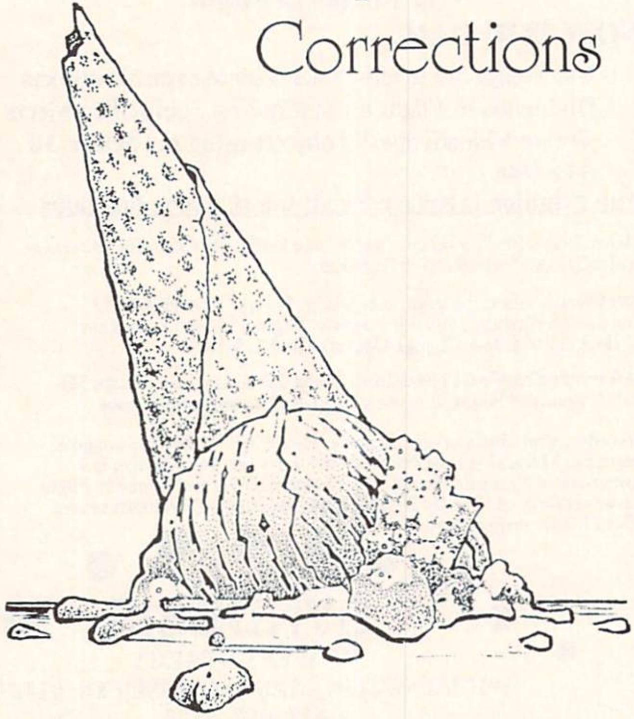
```
ItoHA1_             ;BASIC Fn: A$ = STR1HEX$(I)
                    ;          (A0)         (D0)

   MOVEM.L D1,-(A7);(Push registers)

   MOVEQ.L #1,D1    ;Set the field width to 1
   JSR     ItoHA_   ;Convert the field

   MOVEM.L (A7)+,D1;(Pop registers)

   RTS              ;Return


ItoHA_              ;BASIC Fn: A$ = STRHEX$(I)
                    ;          (A0)         (D0)

;In: D1 = the number of hex-ASCII characters to
;         convert

   MOVEM.L D0-D5,-(A7) ;(Push registers)

   MOVE.L  D1,D2    ;(Set D2 to the field width -1)
   SUBQ.L  #1,D2
   MOVE.L  (A0)+,D3;If the max length of A$ < the
                    ; field width,
   CMP.L   D1,D3    ; (Make A0 point to the current
                    ; useage of A$)
   BLT     ItoHA_7 ; then jump to ItoHA_7
                    ; (Set D3 to the max length of
                    ; A$)
   MOVE.L  D1,(A0)+;Set the current useage of A$
                    ; to the field width
                    ;(Make A0 point to the first
                    ; data byte of A$)
   CMPI.L  #1,D1    ;If the field width is 1,
   BEQ     ItoHA_2 ; then jump to ItoHA_2
```

(continued)

```
   MOVE.L  D2,D4   ;Rotate I right by the field
   SUBQ.L  #1,D4   ; width -1 times 4 bits
ItoHA_1
   ROR.L   #4,D0
   DBRA    D4,ItoHA_1

ItoHA_2
   MOVE.L  D2,D4   ;Set the counter to the field
                   ; width -1
ItoHA_3
   MOVE.B  D0,D5   ;Convert I bits 3-0 to a
   AND.L   #$0F,D5 ; hex-ASCII character
   ADD.L   #'0',D5
   CMP.L   #'9'+1,D5
   BLT     ItoHA_4
   ADDQ.L  #7,D5
ItoHA_4
   MOVE.B  D5,(A0)+;Place the character in A$
                   ;(Make A0 point to the next
                   ; data byte in A$)
   ROL.L   #4,D0   ;Rotate I left by 4 bits
   DBRA    D4,ItoHA_3;Decrement the counter
                   ;If the counter is not yet -1,
                   ; then jump to ItoHA_2
   BRA     ItoHA_9 ;Jump to ItoHA_9

ItoHA_7
   MOVE.L  D3,(A0)+;Set the current useage of A$
                   ; to the max length of A$
   SUBQ.L  #1,D3   ;Fill A$ with astericks
ItoHA_8
   MOVE.B  #'*',(A0)+
   DBRA    D3,ItoHA_8

ItoHA_9
   MOVEM.L (A7)+,D0-D5 ;(Pop registers)

   RTS             ;Return


IsHexASCII              ;Is hex-ASCII routine

;This routine determines whether or not a
; character is a valid hex-ASCII character

;In: <D0.B> = the character

;Out: <Zero> = Clear if the character is a
;              hex-ASCII character,
;            = Set if the character is not a hex-
;              ASCII character
   MOVE.L  D0,-(A7)     ;Push registers

   ANDI.L  #$FF,D0      ;Strip off all but bits 7-0

   CMPI.B  #'0',D0      ;If the character is not a
                        ; hex-ASCII character,
   BMI     IsHex8       ; then jump to IsHex8
   CMPI.B  #'9'+1,D0    ;If in the range "0" - "9",
   BMI     IsHex4       ; then jump to IsHex4
   CMPI.B  #'A',D0
   BMI     IsHex8
   CMPI.B  #'F'+1,D0    ;If in the range "A" - "F",
   BMI     IsHex4       ; then jump to IsHex4
   CMPI.B  #'a',D0
   BMI     IsHex8
   CMPI.B  #'f'+1,D0    ;If in the range "a" - "f",
   BMI     IsHex4       ; then jump to IsHex4
   BRA     IsHex8       ;Jump to IsHex8

IsHex4
   MOVE.L  (A7)+,D0     ;Pop registers
   ANDI    #$FB,CCR     ;Clear the zero flag
   BRA     IsHex9
IsHex8
   MOVE.L  (A7)+,D0     ;Pop registers
   ORI     #$04,CCR     ;Set the zero flag
IsHex9
   RTS                  ;Return
```

•AC•

Oops . . . Corrections



No matter how carefully a project is mapped out, errors always seem to make their way between the bindings. Our on-going Amiga Buyer's Guide is no exception. The Amiga Audio Products guide, listed on pages 70-71 of AC V3.4, suffered from two errors. The following products by Sound Quest, Inc. were omitted:

DXII Master, D50 Master, MT32 Master, TX81Z Master, DX Master, SQ80 Master, CZ Master, Generic Master
$125-$175
Editor/libarians for various indicated synthesizers.

Sound Quest, Inc.
5 Glenaden Ave.
Toronto, Canada M8Y 2L2

Perfect Sound by MicroSearch, Inc. was mistakenly named as a software product.

Perfect Sound is, in fact, hardware— an audio digitizer that digitizes from any line-level input source (such as a stereo, VCR, compact disk player, or amplified microphone) and can create IFF instruments for use with most music programs. The hardware comes complete with all software and a library of pre-recorded sounds.

We apoligize for any inconvenience caused by our error and encourage you to inform us of any missing or inaccurate info in any installments of our Buyer's Guide.

Customizing the Amiga with...

The Companion

...a project that lets you open doors into the Amiga's Event Handling capability.

by Paul Gosselin
PLINK: OLS389 — CompuServe (73117,636)

One day in August, I was sitting back in my chair, hacking away at a program that I was sure would one day make me rich and famous, when I needed to use the mouse. I stretched, but it was three inches beyond my reach. After a few grunts and groans, I sat up and used Mr. Mouse. As I sat back in my chair, I remembered the Amiga could be used without the mouse for many applications, so I decided to try it.

Well, if you've tried it, you know using the Amiga and Shift keys with the cursor keys, and then ALT to simulate the mouse buttons can be a little awkward. After trying that once or twice, I decided to do something to remedy this problem (many inventions came about because of laziness). I mean, I *am* a programmer. I should do something to customize the machine so it works better for *me*.

That's how this project, code-named "The Companion," came about. I decided to write a program that would let me use the cursor keys to look at the Menu of a given window with greater ease than the current method allowed.

I started by looking at the *Rom Kernel Manual* and the source code for an older version of PopCLI, a Public Domain program that uses "hot keys" in much the same way I wished to. Unfortunately, the version of PopCLI I owned used a lot of assembly language. Since I didn't relish the thought of getting wrapped up in Assembly Code, I took a second look at the *Rom Kernel Manual.*

There I found an interesting example program that would place an event handler (of one's own making) in the input stream *before* Intuition could figure out what it meant. This was just what the doctor ordered! If I just typed in that example, I could see events as they came in, and better yet, I could change the program and maybe modify the information. Right? Wrong! Well, at least at first.

After typing the example (along with the tiny assembly part, which was no big deal), I got everything to compile. But it just wouldn't link! Something was missing. Closer examination revealed that some necessary timer-related functions were not present in the code I typed in. Where were they? Well, as the *RKM* stated, the missing functions were located in an example in the Timer section. Examination of the Timer section revealed that the functions were there in spirit, but not in name or syntax. After a little tinkering, and some thought concerning a section that had a few lines out of order, I got the program to work the way it was supposed to.

The next step in the project was a sort of sidestep. To view the Menu properly, I would first have to learn about the Menus and their structures. I discovered I could go about finding the Menu Structure for the active window, and then "walk" through the menu structure. Listing One shows how you can access just about all of the necessary information, and print it out just to make sure you got it. When you use the program, I would suggest you start the program you want to investigate (with the "run" command), and then perform a "wait 4" followed immediately with execution of the menu search program. Before the wait is complete, you should have enough time to click inside the desired window. Redirection is also possible, and is suggested for long menu trees.

With that behind me, it was time to create. The first version of this project simply looked down through the Menu tree and expected that NextMenu or NextItem would give the next menu or item branch. Although this worked in theory, it did not always work in reality.

Several programs yielded humorous results. For example, pressing Right Arrow makes you move left, Down arrow makes you go up, and vice versa. You have to search though the current branch of the menu, and find what really is the next menu selection for appropriate Cursor presses.

THE LISTINGS

Listing Two is the result of this effort. Listing Three is the small machine language section from the *RKM*. The setup of Listing Two, as well as the revised timer functions, are also from the *RKM*.

(continued)

ATTENTION!
PROGRAMMERS & DEVELOPERS
The GS-1000
Graphics Library

★ A full featured INTERACTIVE Graphics Library maximizing the Amiga's™ powerful graphics capabilities!
★ One word commands eliminate hours and hours of programming time!
★ A complete library at your disposal, extensively documented and ready to use! Over 90 commands in all!
★ Built-in features include:

Device Independent 2D Graphics • Multiple Coordinate Representations • Multiple Viewports and Screens • Zoom • Panning • Axial Scaling and Rotation • Rays • Palettes • Undo System w/Get Slate - Put Slate features • Color Mixing • Polymarkers • Brushes • Linestyles • Clipping • Interactive Text I/O • Debug Reporting facilities • PLUS standard features too extensive to list

PLUS A full featured Paint Program w/source to demonstrate the power of the Interactive Library.

Programmed in C, requires 512K memory.
Specifiy "C" Compiler when ordering.
"Amiga is a registered trademark of Commodore-Amiga, Inc."

SEND CHECK OR MONEY ORDER TO:
Rittinghouse Software Development Co.
RR 2, Box 62, Parker, SD 57053

ONLY $49.95
w/source $149.95

VISA/MasterCard Customers Call:
Computer Business Center
(605) 342-6229 • Mon.-Fri. 9-5, Sat. 10-3
DEALERS INQUIRIES INVITED. Call for more info.

The major functions in Listing Two are:

main()
handles setup and takedown of the input handler.

myhandler()
gets events & decides what to do with them.

SNP()
sets new position in the menu, depending on where the pointer currently resides.

go()
figures out where the pointer should go, depending on where it is and the dimensions of the menu.

IN DETAIL

The main routine, taken from the *ROM Kernel Manual*, goes through the usual process to get things started. The IntuitionBase Library is opened, the program sets its priority to twenty to avoid delays in processing, the input.device is opened, and the handler is added to the input stream. A priority greater than fifty (that of Intuition) assures that my handler will get the information before Intuition does, so I can modify incoming signals before Intuition acts on them.

Next is a forever loop which simply loops at a specified interval (via the WaitForTimer function), waiting for the user to type the key that disables the new input handler. When the special key is pressed, the program breaks out of the forever loop, removes the special handler, and exits gracefully. For my purposes, I changed the original main function to modify what went on in the forever loop. Listing #4 contains a working version of the *ROM Kernel Manual* event handler example that inspired this approach.

THE HANDLER

The first argument to the handler function (myhandler) is a linked list of actual raw input events. These events can include, among other things, disk insertion/removal, key presses, and mouse movements. (The original program simply passed this chain of events to Intuition without modification, but first copied all pertinent data to a "dummy" event structure.) Meanwhile, the forever loop prints out the last copied input event data, and if it notices that the F1 key has been pressed (and released), it exits, removing the handler as it goes. Since Timer events can happen so often, they are noted and passed directly to Intuition to avoid unnecessary delays.

My handler, like the original handler, returns immediately if the first event is a timer event. The event data is stored in the structure called "copyevent" which is checked by the forever loop in the main function. Also, a Forbid() call is made to suspend multi-tasking while changing (or viewing) crucial memory structures. After the top event has been copied, the handler scans the linked list for events of interest. Table 1 shows a list (from the include files) of the possible Input Event Classes (IECLASS). If the event was *not* a RAWKEY event (i.e., not a keyboard event), then the handler quickly goes on to the next event without significant delay. If no RAWKEY events are found, the handler makes sure to return the linked list of events to the next handler in the stream (Intuition, most likely).

Another way of speeding the processing uses the switch/case functions to check the Input Event Code values (IECODE; see Table 2), which represent the actual key pressed from the keyboard matrix (see Figure 1 for the A1000 keyboard RAWKEY codes). For easy programming and legibility, #defines were used. The first possibility accepted by the handler is the reverse apostrophe key (just above TAB). This is the hot key that lets you browse among the menus, using the cursor keys. The first thing checked is the Input Event Qualifier (IEQUALIFIER; see Table 3). This tells which other things, if any, may have been going on during the hot-key press.

The test for "LCOMMAND" ensures that the left Amiga key was being held down while the hot key was pressed, to keep the user from getting menus by accident. (It would be a mistake to change the function of a standard keyboard key.) Once the handler is satisfied that the user wants to browse the menu structure from the keyboard (remember, the mouse still functions), the handler changes the event from a RAWKEY press to pressing down the right mouse button. (Intuition assumes the button is held until a new event tells otherwise.) In addition, a dummy event is created to fool Intuition into thinking that the mouse had been moved. How does Intuition know where to move the mouse? Well, a function called "go" takes care of that, and will be described a little later.

Before moving the mouse, the handler stores away special information, such as the mouse's original location. It also brings the pointer to the first Menu structure to be browsed. Note that if the "go" function encounters an error while moving to the HOME position (arbitrarily located at the center of the first Menu header), the handler assumes that there are no menus associated with the current window, and passes the keypress to Intuition.

The next character that might be pressed is the RETURN key. This key tells the handler that the highlighted menu item is to be selected (the equivalent of releasing the right mouse button). Since we are working with Intuition, this event is all that is required to make the proper menu selection. Another dummy event, coupled with the button release, simulates the movement of the mouse back to where it had been when the hot key was selected.

The next presses of interest are the cursor keys. When the browsing mode is established, pressing cursor keys tells the handler to move the mouse to a new position (if possible). After returning from the "go" function, the X and Y values for any mouse movements are stored in the "dummyevent" and "tempEvent" structures which are passed on to Intuition.

After exhausting all possible key presses of interest, the handler returns the pointer to the modified linked list of events to the next handler in the input stream. This could be one of your own design, a program like PopCLI, or Intuition itself.

GOING PLACES

Movement of the mouse requires a call to the routine "go." Go takes two arguments: where and how. Where is HOME, UPARROW, DOWNARROW, LEFTARROW, or RIGHTARROW. How determines if one of the shift keys has been pressed.

(continued)

TxEd *Plus...* All the speed and simplicity of the original TxEd, *Plus...*

Modular software is coming. With modular software, different programs talk to each other using a common macro language. Apple Computer, Inc. has started a software division focusing on modular software, and Microsoft Inc's Bill Gates has been talking about it. Modular software lets "multitasking" mean more than just running two programs at the same time; you can run programs *together,* doing more than each program can do alone and letting you pick exactly the pieces you want to use.

On the Amiga, modular applications are more than just next year's dream. They're available now, with two of the cornerstones ready to go. TxEd *Plus*, the text editor, and AREXX, the macro processor.

Even without the AREXX connection, TxEd *Plus* is the text editor of choice for the Amiga. With AREXX, TxEd *Plus* becomes more than just a text editor. The configurable menus allow you to create customized applications such as order entry systems, and that's barely scratching the surface of the possibilities.
We don't even know what the limits are yet.

Developers: find out about AREXX! The window is wide open.
Users: demand AREXX capability in your software. You can get it now on your Amiga, or wait till next year, on a McClone.

○ Fully configurable menus and keyboard
○ Powerful command line language, uses the AREXX macro processor.
○ Includes functional AREXX demo
○ Many other new features.
○ Uses ARP V1.1

TxEd *Plus* $79.95
TxEd
Microsmiths, Inc
PO Box 561, Cambridge, MA 02140
(617) 354-1224
BIX: cheath CIS: 74216,2117

Mass Residents add 5%. Visa & MC accepted.
Amiga is a trademark of Commodore Business Machines, Inc.
The term 'McClone' is a fictitious conglomeration refering to nothing in particular.

MOVING?

SUBSCRIPTION PROBLEMS?

Please don't forget to let us know.
If you are having a problem with your subscription or if you are planning to move, please write to:

Amazing Computing Subscription Questions
PiM Publications, Inc.
P.O. Box 869
Fall River, MA 02722

Please remember, we cannot mail your magazine to you if we do not know where you are.

Please allow four to six weeks for processing.

The first order of business is finding out where the mouse is. If the mouse is sitting atop any available menu item or subitem, this section will find it and set all of the variables for the selected menu item. From here, the program can figure out where to go when a cursor key is pressed. Unfortunately, this function uses a lot of local variables. Generally, variables starting with "a" are associated with the Menu Header; "b" corresponds to actual Menu Items; "c" is for Menu SubItems. For example, variable "c" is the selected SubItem in the linked list, "cup" tells where to go in the linked list if the "UPARROW" key is pressed, and "cdn" tells where to go if "DOWNARROW" is pressed.

The lowest mathematical position (i.e. the highest on the screen) available in the SubItem linked list is "clow," and is selected if a SHIFTED UPARROW is pressed. In the special cases where an Item has SubItems, I chose to set one of 4 possible flags to note possible movement, rather than create another bunch of variables for right and left. The variable "stoir" is set if pressing the RIGHTARROW results in movement from a SubItem to an Item. The variable "itosr" is set if RIGHTARROW is a valid movement from an Item to a SubItem. As it's coded, the routine knows the difference between SubItems placed to the left *or* right of the respective Item, making Menu movement less complicated.

TABLE ONE

Event Classes

IECLASS_NULL	0x00
IECLASS_RAWKEY	0x01
IECLASS_RAWMOUSE	0x02
IECLASS_EVENT	0x03
IECLASS_POINTERPOS	0x04
IECLASS_TIMER	0x06
IECLASS_GADGETDOWN	0x07
IECLASS_GADGETUP	0x08
IECLASS_REQUESTER	0x09
IECLASS_MENULIST	0x0A
IECLASS_CLOSEWINDOW	0x0B
IECLASS_SIZEWINDOW	0x0C
IECLASS_REFRESHWINDOW	0x0D
IECLASS_NEWPREFS	0x0E
IECLASS_DISKREMOVED	0x0F
IECLASS_DISKINSERTED	0x10
IECLASS_ACTIVEWINDOW	0x11
IECLASS_INACTIVEWINDOW	0x12

TABLE TWO

Event Codes

IECODE_UP_PREFIX	0x80
IECODE_KEY_CODE_FIRST	0x00
IECODE_KEY_CODE_LAST	0x77
IECODE_COMM_CODE_FIRST	0x78
IECODE_COMM_CODE_LAST	0x7F
IECODE_C0_FIRST	0x00
IECODE_C0_LAST	0x1F
IECODE_ASCII_FIRST	0x20
IECODE_ASCII_LAST	0x7E
IECODE_ASCII_DEL	0x7F
IECODE_C1_FIRST	0x80
IECODE_C1_LAST	0x9F
IECODE_LATIN1_FIRST	0xA0
IECODE_LATIN1_LAST	0xFF
IECODE_LBUTTON	0x68
IECODE_RBUTTON	0x69
IECODE_MBUTTON	0x6A
IECODE_NOBUTTON	0xFF
IECODE_NEWACTIVE	0x01
IECODE_REQSET	0x01
IECODE_REQCLEAR	0x00

TABLE THREE

Event Qualifiers

IEQUALIFIER_LSHIFT	0x0001
IEQUALIFIER_RSHIFT	0x0002
IEQUALIFIER_CAPSLOCK	0x0004
IEQUALIFIER_CONTROL	0x0008
IEQUALIFIER_LALT	0x0010
IEQUALIFIER_RALT	0x0020
IEQUALIFIER_LCOMMAND	0x0040
IEQUALIFIER_RCOMMAND	0x0080
IEQUALIFIER_NUMERICPAD	0x0100
IEQUALIFIER_REPEAT	0x0200
IEQUALIFIER_INTERRUPT	0x0400
IEQUALIFIER_MULTIBROADCAST	0x0800
IEQUALIFIER_MIDBUTTON	0x1000
IEQUALIFIER_RBUTTON	0x2000
IEQUALIFIER_LEFTBUTTON	0x4000
IEQUALIFIER_RELATIVEMOUSE	0x8000

COMPUTER MART
Your Texas Amiga Source
Immediate Access to over 400 Amiga Titles.
Prices too low to print!

We stock Amiga Software and Peripherals
For A500, A1000, & A2000.
Mon. thru Fri. 10:00 AM-7:00 PM, Sat. 12:00-5:00 PM

CALL TOLL FREE
800-443-8236

CUSTOMER SERVICE
409-560-2826

Computer Mart . 105 Lynn Street . Nacogdoches, TX 75961

Once the possible movements have been calculated, another switch/case combination decides how a call to the SetNewPosition function will be made. The SNP function calculates where to place the mouse pointer. The parameters indicate the depth of the Header, Item, and SubItem of the entire Menu tree in the limited lists. Note how the use of #defines makes reading easier. If not shifted, the mouse moves to the near item, but a shifted keypress results in a move to an extreme position in the menu strip.

THE SNP FUNCTION

The SNP function uses the characteristics of the menu structure to calculate where the mouse must move. Because some programs have SubItem lists that overlap more than half of the Item containing the SubItems, a temporary Event "tempEvent" was created for the linked list. Its function is to zip the mouse to the header, and to make Intuition remove the SubItem list, enabling the mouse to be properly placed on the next Menu Item as positioned by the "dummyEvent" Event. ScaleX and ScaleY were needed to handle the various possible screen resolutions. Although the means of determination (ScreenHeight > 202 and ScreenWidth > 322) are by no means the best, they do the job quite well.

COMPILING

Assemble listing three with any assembler which allows linking with C object files. I compiled the C part, using Lattice 3.1, as follows:

```
lc -d -v -r -s companion
 Then link:
   Blink FROM lib:c.o comp.o companion.o to companion
         LIB lib:lc.lib+lib:amiga.lib
         NODEBUG VERBOSE SMALLCODE SMALLDATA
```

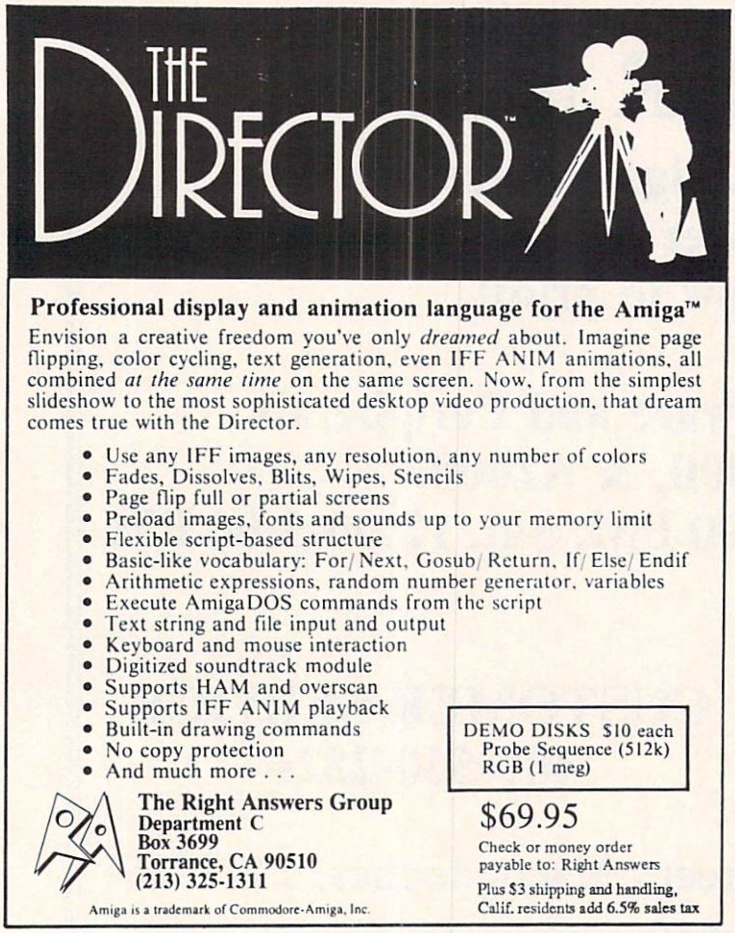
where comp.o is the output of the assembler.

PROGRAM USE

To use "The Companion," simply run the program from CLI, or click the Icon from Workbench. (You'll need to create your own Icon for it.) To keep Workbench from opening up a dummy window, the function "main" was renamed " main," which skips over the setting of stdio and doesn't open the workbench window.

Holding the left Amiga key while pressing the reverse apostrophe (') key should make it appear as if the right mouse button is being held down and the mouse is highlighting the left menu header. The cursor keys should allow movement

(continued)


THE DIRECTOR™

Professional display and animation language for the Amiga™
Envision a creative freedom you've only *dreamed* about. Imagine page flipping, color cycling, text generation, even IFF ANIM animations, all combined *at the same time* on the same screen. Now, from the simplest slideshow to the most sophisticated desktop video production, that dream comes true with the Director.

- Use any IFF images, any resolution, any number of colors
- Fades, Dissolves, Blits, Wipes, Stencils
- Page flip full or partial screens
- Preload images, fonts and sounds up to your memory limit
- Flexible script-based structure
- Basic-like vocabulary: For/Next, Gosub/Return, If/Else/Endif
- Arithmetic expressions, random number generator, variables
- Execute AmigaDOS commands from the script
- Text string and file input and output
- Keyboard and mouse interaction
- Digitized soundtrack module
- Supports HAM and overscan
- Supports IFF ANIM playback
- Built-in drawing commands
- No copy protection
- And much more . . .

DEMO DISKS $10 each
Probe Sequence (512k)
RGB (1 meg)

The Right Answers Group
Department C
Box 3699
Torrance, CA 90510
(213) 325-1311

$69.95

Check or money order
payable to: Right Answers

Plus $3 shipping and handling,
Calif. residents add 6.5% sales tax

Amiga is a trademark of Commodore-Amiga, Inc.

throughout most of the menu structure, jumping over ghosted items and moving to the extreme positions when the shift is held with the cursor key.

BUGS

The program is actually crash-free, but it does have one shortcoming. To make coding simpler for this example, left and right arrow key presses are assumed to be meaningful when moving along the header, and between Items and SubItems. Some application programs have "two-dimensional" menu structures, and this program may not find some of the possible items if they have the same TopEdge value. Since very few programs use this, the mouse can be used to move over to the unreachable items and a press of the Return key will still register the selection. This can be corrected by a more extensive search through the menu list.

GOING FURTHER

By using the basic ideas shown here, anyone can customize the Amiga to suit their needs. This technique could be used in turn-key applications to easily utilize the powerful Amiga Menu system, where a mouse is not desired. Similar techniques can be used for answering requesters or pressing gadgets. It beats the built-in method by leaps and bounds. Who knows, maybe we can even see those secret messages mentioned in *Amazing Computing* V2.8, without needing four fingers, two toes, and two noses (while inserting a disk in drive 0).

By the way, that program I mentioned at the start of the article that would make me rich and famous has long since been forgotten.

Listing One: Men.c

```
#include <stdio.h>
#include <intuition/intuitionbase.h>
extern struct IntuitionBase *IntuitionBase;

main()
{
   struct Menu *menu;
   struct MenuItem *item;
   struct MenuItem *subitem;
   char *text;
   struct IntuiText *itemfill;
   /* Open IntuitionBase, Exit if fails */
   IntuitionBase-(struct IntuitionBase
      *)OpenLibrary(“intuition.library”,-1);
   if(IntuitionBase -- NULL)
   {
      printf(“Couldn’t open IntuitionBase\n”);
      exit(20);
   }

   /* Grab onto the First menu of the currently active window */
   menu - (struct Menu *)IntuitionBase->ActiveWindow->MenuStrip;
   while( menu !- NULL )  /* while there are more (any) menus */
   {
      text - menu->MenuName; /* Grab onto Menu text */
      printf(“%s\n”,text);
      item - menu->FirstItem;/* Grab onto first menu item */
      while( item !- NULL )   /* while there are more (any) items */
      {
         /* itemfill can be either Image or IntuiText structures */
         if( (item->Flags & ITEMTEXT ) )  /* Text */
         {
            itemfill - (struct IntuiText *)item->ItemFill;
            text - itemfill->IText;
            printf(“   %s\n”,text);
         }
         else /* Image */
         {
            printf(“    IMAGE DATA\n”);
         }
         subitem - item->SubItem;
         while( subitem !- NULL )
         {
            /* itemfill can be either Image or IntuiText structures */
            if( (subitem->Flags & ITEMTEXT ) )       /* Text */
            {
               itemfill - (struct IntuiText *)subitem->ItemFill;
               text - itemfill->IText;
               printf(“        %s\n”,text);
            }
            else /* Image */
            {
               printf(“        IMAGE DATA\n”);
            }
            subitem - subitem->NextItem;/* Get next subitem */
         }
         item - item->NextItem; /* Get next item */
      }
      menu - menu->NextMenu; /* Get next menu */
   }
   /* cleanup */
   CloseLibrary(IntuitionBase);
   return(0);
}
```

Listing Two: Companion.c

```
#include <exec/types.h>
#include <exec/ports.h>
#include <exec/memory.h>
#include <exec/io.h>
#include <exec/tasks.h>
#include <exec/interrupts.h>
#include <devices/input.h>
#include <exec/devices.h>
#include <devices/inputevent.h>
```

```
#include <intuition/intuitionbase.h>
extern struct IntuitionBase *IntuitionBase;

struct InputEvent copyevent;
struct MsgPort *inputDevPort;
struct IOStdReq *inputRequestBlock;
struct Interrupt handlerStuff;
struct InputEvent dummyEvent,tempEvent;

extern struct MsgPort *CreatePort();
extern struct IOStdReq *CreateStdIO();

struct MemEntry me[10];

/* My special defines and variables */
#define HOME    0xff
#define UPARROW 0x4c
#define DOWNARROW 0x4d
#define RIGHTARROW0x4e
#define LEFTARROW 0x4f
#define TOP     0x01
#define ITEM    0x02
#define SUBITEM 0x03
#define FASTMOUSE 0x3f
#define SLOWMOUSE 0x3d
#define IAS IntuitionBase->ActiveScreen
#define IAW IntuitionBase->ActiveWindow
#define SUB_NOT_GHOSTED ( (C_subitem->Flags & ITEMENABLED)!=0 )
#define ITM_NOT_GHOSTED ( (C_item->Flags & ITEMENABLED)!=0 )
#define T_ITM_NOT_GHOSTED ((T_item->Flags & ITEMENABLED)!=0 )
#define MEN_NOT_GHOSTED ( (C_header->Flags & MENUENABLED)!=0 )
#define T_MEN_NOT_GHOSTED ((T_header->Flags & MENUENABLED)!=0 )
#define SHIFTED_MOVE (
(how==IEQUALIFIER_LSHIFT)||(how==IEQUALIFIER_RSHIFT) )

int SaveX,SaveY;  /* Place to save mouse position when "browsing" */
int browsing = 0;

struct Menu *currenttop;
struct MenuItem *currentitem,*From_item;
struct MenuItem *currentsubitem;

int Ta,Tb,Tc,error,TotX=0,TotY=0;
int ScaleX=2, ScaleY=2;/* MouseMove event scaling factor */
/* End of my stuff */

struct InputEvent
*myhandler(ev,mydata)
struct InputEvent *ev;
struct MemEntry *mydata[];
{
   register struct InputEvent *ep, *laste;
   if(ev->ie_Class==IECLASS_TIMER)
   {
      return(ev);
   }
   else
   {
      Forbid(); /* Copy first event for "main" roiutine */
      copyevent.ie_Class = ev->ie_Class;
      copyevent.ie_SubClass = ev->ie_SubClass;
      copyevent.ie_Code = ev->ie_Code;
      copyevent.ie_Qualifier = ev->ie_Qualifier;
      copyevent.ie_X = ev->ie_X;
      copyevent.ie_Y = ev->ie_Y;
      copyevent.ie_TimeStamp.tv_secs = ev->ie_TimeStamp.tv_secs;
      copyevent.ie_TimeStamp.tv_micro = ev->ie_TimeStamp.tv_micro;

      for (ep = ev, laste = NULL; ep != NULL; ep = ep->ie_NextEvent )
      {
         if( ep->ie_Class == IECLASS_RAWKEY )
         {
            switch( ep->ie_Code )
            {
               case 0x00 :
               if( ep->ie_Qualifier == (IEQUALIFIER_LCOMMAND |
                  IEQUALIFIER_RELATIVEMOUSE) )
               {
                  Ta=1; Tb=0 ; Tc=0;
                  if( error=go(HOME,0x00) ) { Permit(); return(ev); }

                  ep->ie_Class = IECLASS_RAWMOUSE;
                  ep->ie_Code  = IECODE_RBUTTON;
                  ep->ie_Qualifier = IEQUALIFIER_RBUTTON;
                  ep->ie_X     = 0;
                  ep->ie_Y     = 0;
                  SaveX=IntuitionBase->ActiveScreen->MouseX;
                  SaveY=IntuitionBase->ActiveScreen->MouseY;
                  currenttop = (struct Menu *) IntuitionBase->
                     ActiveWindow->MenuStrip;
                  if( currenttop == NULL) { Permit(); return(ev); }
                  browsing=1;
                  dummyEvent.ie_Class = IECLASS_RAWMOUSE;
                  dummyEvent.ie_Code = IECODE_NOBUTTON;
                  dummyEvent.ie_Qualifier = IEQUALIFIER_RELATIVEMOUSE;
                  dummyEvent.ie_NextEvent = ep->ie_NextEvent;
                  ep->ie_NextEvent = &dummyEvent;
```

```
               }
               break;
               case 0x44 :
               if( browsing==1  )
               {
                  ep->ie_Class = IECLASS_RAWMOUSE;
                  ep->ie_Code  = IECODE_RBUTTON | IECODE_UP_PREFIX;
                  ep->ie_Qualifier = IEQUALIFIER_RBUTTON;
                  ep->ie_X     = 0;
                  ep->ie_Y     = 0;
                  browsing=0;
                  dummyEvent.ie_Class = IECLASS_RAWMOUSE;
                  dummyEvent.ie_Code = IECODE_NOBUTTON;
                  dummyEvent.ie_Qualifier = IEQUALIFIER_RELATIVEMOUSE;
                  dummyEvent.ie_X = ScaleX*(SaveX -
                        IntuitionBase->ActiveScreen->MouseX);
                  dummyEvent.ie_Y = ScaleY*(SaveY -
                        IntuitionBase->ActiveScreen->MouseY);
                  dummyEvent.ie_NextEvent = ep->ie_NextEvent;
                  ep->ie_NextEvent = &dummyEvent;
               }
               break;
               case DOWNARROW :

               case UPARROW :

               case LEFTARROW :

               case RIGHTARROW :
               if(browsing == 1)
               {
                  error=go(ep->ie_Code,ep->ie_Qualifier);
                  if( error ) { Permit(); return(ev); }

               /* tempEvent makes pointer momentarily jump to */
               /* Header before actually moving for those times*/
               /* that subitems "get in the way" */

                  tempEvent.ie_Class = IECLASS_RAWMOUSE;
                  tempEvent.ie_Code = IECODE_NOBUTTON;
                  tempEvent.ie_Qualifier = IEQUALIFIER_RELATIVEMOUSE;
                  tempEvent.ie_NextEvent = &dummyEvent;


                  dummyEvent.ie_Class = IECLASS_RAWMOUSE;
                  dummyEvent.ie_Code = IECODE_NOBUTTON;
```

(continued)

AMIGA DUAL 3½" DISK DRIVES

100% Compatable with Amiga 500, 1000 & 2000 Computers

- **Internal Power Supply**
- **All Metal Chassis**
- **Horizontal Layout**
- **Vented Enclosure**
- **On-Off Switch**
- **6 foot 3-prong linecord**
- **Primary Circuit Breaker Protected**
- **Color Coordinated to Amiga Computers**

ONLY $395.00

Single Disk Drive w/pass-thru **$199.00**

20 Meg Hard Drive (SCSI) with Controller

Only $699.00

Comp-U-Save

414 Maple Avenue, Westbury, NY 11590
In NY State (516) 997-6707
Outside NY State (800) 356-9997

Call for our complete PD Software catalog.

JUMPDISK
The Double

JUMPDISK is the original disk magazine for the Amiga®. Every month for 22 months: NEW ready-to-run programs, articles, tips, reviews, art and music.

- Get the current issue for $9.
- Or for just $3, try the JUMPDISK SAMPLER.
- Best of all, order the $10 DOUBLE, both the current issue and the Sampler. Two disks. Crammed.

IF YOU DON'T LIKE WHAT WE SEND, RETURN IT FOR FULL REFUND.

Order from:

JUMPDISK
1493 Mt. View Ave.
Chico, CA 95926

- Not convinced? Get our Big Emotional Pitch mailed to you by either writing us at the address above, or calling us at (916) 343-7658.
- We ship day order is received. Personal check OK.
- Amiga® is a registered trademark of Commodore-Amiga.
- Look, Ma, I'm programming!

```
                dummyEvent.ie_Qualifier = IEQUALIFIER_RELATIVEMOUSE;
                dummyEvent.ie_NextEvent = ep->ie_NextEvent;

                ep->ie_NextEvent = &tempEvent;
             }
             break;
             default :
             break;
          }  /* End of Switch/case */
       }  /* End of if RAWKEY */
       else
       laste = ep;

    }  /* End of for loop */

    Permit();
  }   /* end of original if/else */
  return(ev);
}


int SNP(header,item,subitem)
int header,item,subitem;
{
  int BarHeight,TemX,TemY;
  int CmouseX,CmouseY,a,b,c;
  struct Menu *T_header;
  struct MenuItem *T_item,*T_subitem;

  CmouseX=IAS->MouseX; /* Current mouse positions */
  CmouseY=IAS->MouseY;

  dummyEvent.ie_X=0;   /* Clear dummyEvent position variables */
  dummyEvent.ie_Y=0;

  T_header = (struct Menu *)IAW->MenuStrip; /* Get current MenuStrip */
  for(a=1;a<header && T_header!=NULL ;a++,T_header
       = (struct Menu *)T_header->NextMenu);
  /* Found desired Menu Header */


  T_item = (struct MenuItem *)T_header->FirstItem; /* Get first Item */
  for(b=0;b<item-1 && T_item!=NULL  ;b++,T_item
       = (struct MenuItem *)T_item->NextItem);
  /* Found desired Menu Item, if any */

  T_subitem = (struct MenuItem *)T_item->SubItem;
       /* Get first Subitem, if any */
  for(c=0;c<subitem-1 && T_subitem!=NULL ;c++,T_subitem
       = (struct MenuItem *)T_subitem->NextItem);
  /* Found desired Menu SubItem, if any */


  /* Quick and dirty way to determine scaling of mousemove
   * events.
   */
  if( IAS->Width >= 322 ) ScaleX=1; else ScaleX=2;
  if( IAS->Height>= 202 ) ScaleY=1; else ScaleY=2;


  BarHeight = IAS->BarHeight;

  /* tempEvent will be used to temporarily jump up to the
   * menu header if a subitem list is drawn, and in the way;
   * in which case DummyEvent simply moves from header
   * to the desired menu item.
   */

  if( T_header == NULL) return(1); /* If there are NO Menus */

  if( a!=header || (b!=item-1 && item!=0)  ||
       (c!=subitem-1 && subitem!=0) )
  {
    return(1);
  }
  if( subitem !=0 )
  {
    TotX = T_header->LeftEdge+T_subitem->LeftEdge+T_subitem->Width/2;
    TotY = T_header->TopEdge + BarHeight -1
    +T_item->TopEdge+T_subitem->TopEdge+T_subitem->Height/2;
    TemX = CmouseX;
    TemY = CmouseY;
    tempEvent.ie_X = 0;   /* i.e. don't move from Item if */
    tempEvent.ie_Y = 0;   /* you want to move within a subitem */
    dummyEvent.ie_X = (TotX-CmouseX)*ScaleX;
    dummyEvent.ie_Y = (TotY-CmouseY)*ScaleY;
    return(0);
  }

  if( item !=0 )
  {
    TotX = T_header->LeftEdge+T_item->LeftEdge+T_item->Width/2;
    TotY = T_header->TopEdge + (BarHeight-1)
       +T_item->TopEdge + T_item->Height/2;
    if( (From_item->SubItem != NULL) && (From_item != NULL) )
    {
       if( From_item->SubItem->LeftEdge > (T_item->LeftEdge
       + T_item->Width/2) )
       {
          TemX = TotX;
          TemY = TotY;
       }
       else
       {
          TemX = T_header->LeftEdge + T_header->Width/2 ;
          TemY = T_header->TopEdge + BarHeight/2 ;
       }
    }
    else
    {
       TemX=TotX;
       TemY=TotY;
    }
    tempEvent.ie_X = (TemX-CmouseX)*ScaleX;
         /* Aha, here we need to jump */
    tempEvent.ie_Y = (TemY-CmouseY)*ScaleY; /* to header in case of */
                                /* nasty subitems badly placed */

    dummyEvent.ie_X = (TotX-TemX)*ScaleX;
         /* dummyEvent then moves from */
    dummyEvent.ie_Y = (TotY-TemY)*ScaleY;
         /* Header to desired Menu item */

    return(0);
  }

  if( header!=0 )
  {
    TotX = T_header->LeftEdge + T_header->Width/2 ;
    TotY = T_header->TopEdge + BarHeight/2;
    TemX = CmouseX;
    TemY = CmouseY;
    tempEvent.ie_X = 0;  /* no need to jump to header before */
    tempEvent.ie_Y = 0;  /* moving, already there! */
    dummyEvent.ie_X = (TotX-CmouseX)*ScaleX;
    dummyEvent.ie_Y = (TotY-CmouseY)*ScaleY;

  }
  return(0);
}
```

```
int go(there,how)
int there,how;
{
   int CmouseX,CmouseY,a-0,b-0,c-0,This_top,This_left;
   int up,dn,bup,bdn,cup,cdn,low,high,blow,bhigh,clow,chigh,alow,ahigh;
   int stoir,stoil,itosr,itosl, rt,lf,art,alf,htoi;
   struct Menu *T_header,*C_header;
   struct MenuItem *T_item,*T_subitem,*C_item,*C_subitem;
   CmouseX-IAS->MouseX;
   CmouseY-IAS->MouseY;
   T_header - (struct Menu *)IAW->MenuStrip;
   dummyEvent.ie_X-0;
   dummyEvent.ie_Y-0;
   how &- 0x00ff;


   /* Find out which, if any menu/item/subitem are highlighted */

   while(T_header !- NULL)
   {
      a++;
      if( (T_header->Flags & MIDRAWN) -- MIDRAWN ) break;
      T_header - (struct Menu *) T_header->NextMenu;
   } /* Exits "a" and T_header -> Selected Header or NULL */
   if( T_header -- NULL) a-0;

   if(T_header !- NULL)
   {
      T_item - (struct MenuItem *)T_header->FirstItem;
      while(T_item !- NULL)
      {
         b++;
         if( (T_item->Flags & HIGHITEM) — HIGHITEM ) break;
         T_item - (struct MenuItem *) T_item->NextItem;
      /* Exits with b and T_item */
      } /* pointing to selected item, or NULL */
      if( T_item — NULL) b-0;
   }
   From_item - T_item;
   if( (T_header !- NULL) && (T_item !- NULL) )
   {
      T_subitem - (struct MenuItem *)T_item->SubItem;
      while(T_subitem !- NULL)
      {
         c++;
         if( (T_subitem->Flags & HIGHITEM) — HIGHITEM ) break;
         T_subitem - (struct MenuItem *) T_subitem->NextItem;
      /* Exits with c and T_subitem */
      /* pointing tp selected subitem, or NULL */
      }
      if( T_subitem -- NULL) c-0;
   }

         /* END of "find where I am" */


   htoi - 0;     /* htoi means "header to item" movement */
   if( (c--0 && b--0 && a!-0) || (there—HOME) )
               /* Movement on HEADER  */
   {
      This_left - T_header->LeftEdge;
      C_header - IAW->MenuStrip;
      Ta-0;
      low - 10000 ; high - -10000; lf = -10000 ; rt - 10000;
      if( (T_header->FirstItem !- NULL) && T_MEN_NOT_GHOSTED )
      {
         htoi - 1;
         T_item - T_header->FirstItem;
      }

      while( C_header !- NULL)
      {
         Ta++;
         if( MEN_NOT_GHOSTED )
         {
            if( (C_header->LeftEdge < This_left) && (C_header->LeftEdge >
lf) )
            {
               lf - C_header->LeftEdge; alf - Ta;
            }
            if( (C_header->LeftEdge > This_left) && (C_header->LeftEdge <
rt) )
            {
               rt - C_header->LeftEdge; art - Ta;
            }
            if( C_header->LeftEdge < low )
            {
               low - C_header->LeftEdge ; alow - Ta;
            }
            if( C_header->LeftEdge > high )
            {
               high - C_header->LeftEdge ; ahigh - Ta;
            }
         }
         C_header - C_header->NextMenu;
      }

   }

   itosr-0;  /* Item to Subitem using Right cursor */
   itosl-0;  /* Item to Subitem using Left cursor */
   if( (c--0 && b!-0) || htoi--1 )   /* Movement on an ITEM  */
   {
      This_top - T_item->TopEdge;
      C_item - T_header->FirstItem;
      Tb-0;
      up - -1000; dn - 1000; bup - bdn - b;
      low - 1000; high - -1000 ; blow - bhigh - b;
      if( (T_item->SubItem !- NULL) && T_ITM_NOT_GHOSTED )
      {
         if(T_item->LeftEdge <- T_item->SubItem->LeftEdge)
         itosr-1;
         else
         itosl-1;
         T_subitem - T_item->SubItem;
      }

      while( C_item !- NULL)
      {
         Tb++;
         if(ITM_NOT_GHOSTED)
         {
            if( (C_item->TopEdge < This_top) && (C_item->TopEdge > up) )
            {
               up - C_item->TopEdge; bup - Tb;
            }
            if( (C_item->TopEdge > This_top) && (C_item->TopEdge < dn) )
            {
               dn - C_item->TopEdge; bdn - Tb;
            }
            if( C_item->TopEdge < low )
            {
               low - C_item->TopEdge ; blow - Tb;
            }
            if( C_item->TopEdge > high )
            {
               high - C_item->TopEdge ; bhigh - Tb;
            }
         }
         C_item - C_item->NextItem;
      }

   }
```

(continued)

Memory designed with some thought ...

If you're an Amiga 2000 owner, you'd undoubtedly like to add more memory to your system. After all, the reason you chose the A2000 was for the expandability. The question now is "What's the best path for upgrading?". A 2 meg memory expansion card may seem the cheapest way, but it's too easily maxed out. You'd like to be able to expand beyond 2 megs, but the sticker shock of an 8 meg board populated with its minimum configuration of 2 megs makes this route unappealing as well. The Digitronics RC4 Ramcard was designed to solve these problems. The RC4 is auto-configurable as low as 1/2 megabyte, but as your memory requirements gradually grow, so can your memory, in steps of 1/2 meg, all the way up to 4 full megabytes of no wait-state fast memory!

If you're an Amiga 500 owner, adding memory to your system may seem a bit more difficult than simply plugging in an expansion card. Some vendors seem to think it's a good idea to tear open your machine and solder in printed circuit boards, voiding your warranty, overheating the inside of the case, and bringing an already overtaxed A500 power supply to its knees. Digitronics offers a much better solution. The same RC4 Ramcard designed to plug into the A2000 can be plugged into the expansion slot on the side of the A500 when installed in our adapter box. This adapter is a completely enclosed low-profile metal case with its own power supply, which can fit neatly under your external A1010. This allows for a quick and easy way to add up to 4 megabytes of fast memory to your A500. Best of all, if you should ever decide to move up to an A2000, the RC4 ramcard you've been using in your A500 can go along as well.

The RC4 comes completely assembled and tested, and is fully socketed for 4 megs of 256k x 4 DIP DRAMS (0k installed). Also included is a disk containing memory test software, our own version of a recoverable RAM Disk, and other useful utilities.

Call or write for pricing of boards with various amounts of RAM, or for information on our do-it-yourself kits.

Dealer inquiries invited.

P O Box 579 Hatfield, PA 19440
(215) 361-1999

Digitronics

• Amiga is a trademark of Commodore-Amiga Inc.

Hard n *Fast*

You bought your Amiga because it is the most powerful personal computer and installed a hard disk to make it even more powerful. Being a knowledgable computer user, you then set out to backup your data.

You started the program and waited . . . through hours of gronking and disk swapping. It was so tiresome that you put off backing up your disk again. If you haven't already lost irreplaceable data, you know that it will happen sooner or later.

@%&¢%! That's how I feel too. I know the Amiga is better than that. Fortunately I know how to program the Amiga right.

Backup your whole disk, a directory, or just the files that have changed since your last backup. Restore it all or just a file or two that you fat-fingered. It's fast, 10 megabytes in under a half hour! It's easy to use, with menus and all that nice Intuition stuff.

And it's only **$49.95**
It's **Hard** n *Fast*

Don's ComputerWorks
6641 Scott Street
Hollywood, Florida 33024
(305) 966-8179

```
stoir=0;  /* Subitem to Item using Right cursor */
stoil=0;  /* Subitem to Item using Left cursor */

if( c!=0 || itosr==1 || itosl==1 )   /* Movement on a SUBITEM  */
{
  This_top = T_subitem->TopEdge;
  C_subitem = T_item->SubItem;
  Tc=0;
  up = -1000; dn = 1000; cup = cdn = c;
  low = 1000; high = -1000 ; clow = chigh = c;
  while( C_subitem != NULL)
  {
    Tc++;
    if( SUB_NOT_GHOSTED )
    {
  if( (C_subitem->TopEdge < This_top) && (C_subitem->TopEdge > up) )
      {
        up = C_subitem->TopEdge; cup = Tc;
      }
  if( (C_subitem->TopEdge > This_top) && (C_subitem->TopEdge < dn) )
      {
        dn = C_subitem->TopEdge; cdn = Tc;
      }
      if( C_subitem->TopEdge < low )
      {
        low = C_subitem->TopEdge ; clow = Tc;
      }
      if( C_subitem->TopEdge > high )
      {
        high = C_subitem->TopEdge ; chigh = Tc;
      }
    }
    C_subitem = C_subitem->NextItem;
  }
  if( c == clow )
  {
    if( T_item->LeftEdge < T_subitem->LeftEdge)
    stoil = 1;
    else
    stoir = 1;
  }
}

error=1;
switch( there )
{
  case HOME :
  error=SNP(alow,0,0);
  break;

  case DOWNARROW :
  /* Moving DOWN from HEADER */
  if( htoi ==1)
  error = SNP(a,blow,0);
  /* Moving DOWN through ITEMS */
  if(b!=0 && c==0 && a!=0)
  {
    if( SHIFTED_MOVE )
    error = SNP(a,bhigh,c);
    else
    error = SNP(a,bdn,c);
  }
  /* Moving DOWN through SUBITEMS */
  if(b!=0 && c!=0 && a!=0)
  {
    if( SHIFTED_MOVE )
    error = SNP(a,b,chigh);
    else
    error = SNP(a,b,cdn);
  }
  break;

  case UPARROW :
  /* Moving UP through ITEMS */
  if(b!=0 && c==0 && a!=0 )
  {
    if( SHIFTED_MOVE )
    error = SNP(a,blow,c);
    else
    {
      if(b==blow)
      error = SNP(a,0,0);
      else
      error = SNP(a,bup,c);
    }
  }
  /* Moving UP through SUBITEMS */
  if(b!=0 && c>1 && a!=0)
  {
    if( SHIFTED_MOVE )
    error = SNP(a,b,clow);
    else
    error = SNP(a,b,cup);
  }
  break;

  case LEFTARROW :
  /* Moving along header */
  if(a!=0 && b==0 && c==0)
  {
    if( SHIFTED_MOVE )
    error = SNP(alow,0,0);
    else
    error = SNP(alf,0,0);
  }
  /* Moving Back to ITEM from SUBITEM */
  if( stoil == 1)
  error = SNP(a,b,0);
  /* Moving to SUBITEM from ITEM: SPECIAL CASE */
  if( itosl == 1)
  error = SNP(a,b,clow);
  break;
  case RIGHTARROW :
  /* Moving along header */
  if(a!=0 && b==0 && c==0)
  {
    if( SHIFTED_MOVE )
    error = SNP(ahigh,0,0);
    else
    error = SNP(art,0,0);
  }


  /* Moving to SUBITEM from ITEM */
  if( itosr == 1 && b!=0 && c==0 )
  error = SNP(a,b,clow);

  /* Moving to ITEM from SUBITEM: SPECIAL CASE */
  if( stoir == 1 && b!=0 && c!=0 )
  error = SNP(a,b,0);
 }
 TotX=0; TotY=0;
 return( error );
}
extern struct Task *FindTask();
struct Task *mytask;
LONG mysignal;
extern void HandlerInterface();

struct timerequest *mytimerRequest;
extern struct timerequest *PrepareTimer();
extern int WaitForTimer();
extern int DeleteTimer();
```

```
main()
{
   SHORT error;
   ULONG oldseconds,oldmicro,oldclass;
   IntuitionBase-(struct IntuitionBase *)
   OpenLibrary("intuition.library",-1);
   if(IntuitionBase--NULL)
   {
      printf("Couldn't open IntuitionBase\n");
      exit(20);
   }
   dummyEvent.ie_Class - IECLASS_NULL;
   dummyEvent.ie_NextEvent - NULL;
   SetTaskPri( FindTask(0), 20);

   inputDevPort - CreatePort(0,0);
   if(inputDevPort -- NULL)
   {
      printf("Couldn't create Input Port\n");
      CloseLibrary(IntuitionBase);
      exit(-1);
   }
   inputRequestBlock - CreateStdIO(inputDevPort);
   if(inputRequestBlock -- 0)
   {
      printf("Couldn't open inputDevPort \n");
      CloseLibrary(IntuitionBase);
      DeletePort(inputDevPort);
      exit(-2);
   }

   mytimerRequest - PrepareTimer();
   if(mytimerRequest--NULL)
   {
      DeleteStdIO(inputRequestBlock);
      DeletePort(inputDevPort);
      CloseLibrary(IntuitionBase);
      exit(-3);
   }

   handlerStuff.is_Data - (APTR)&me[0];
   handlerStuff.is_Code - HandlerInterface;
   handlerStuff.is_Node.ln_Pri - 51;

   error - OpenDevice("input.device",0,inputRequestBlock,0);
   if(error!-0)
   {
      printf("Couldn't open input.device\n");
      DeleteStdIO(inputRequestBlock);
      DeletePort(inputDevPort);
      DeleteTimer(mytimerRequest);
      CloseLibrary(IntuitionBase);
      exit(-4);
   }

   printf("\n        Amiga Companion\n");
   printf("   LAmiga (`) to enable browse\n");
   printf("   Arrow keys to browse\n");
   printf("   Return to SELECT\n");
   printf("   CTRL (`) to remove companion\n");

   inputRequestBlock->io_Command - IND_ADDHANDLER;
   inputRequestBlock->io_Data - (APTR)&handlerStuff;

   DoIO(inputRequestBlock);
   copyevent.ie_TimeStamp.tv_secs - 0;
   copyevent.ie_TimeStamp.tv_micro - 0;
   copyevent.ie_Class - 0;
   oldseconds-0;
   oldmicro-0;
   oldclass-0;

   for(;;)   /*FOREVER*/
   {
      WaitForTimer(mytimerRequest,0,100000);
      if(copyevent.ie_Class--IECLASS_RAWKEY &&
      copyevent.ie_Code--0x80 &&
      (copyevent.ie_Qualifier & IEQUALIFIER_CONTROL) ) break;

   }
   inputRequestBlock->io_Command - IND_REMHANDLER;
   inputRequestBlock->io_Data - (APTR)&handlerStuff;
   DoIO(inputRequestBlock);

   /* Cleanup */
   CloseDevice(inputRequestBlock);
   DeleteStdIO(inputRequestBlock);
   DeletePort(inputDevPort);
   DeleteTimer(mytimerRequest);
   CloseLibrary(IntuitionBase);
   return(0);
}

struct timerequest *
PrepareTimer()
{
   int error;
   struct MsgPort *timerport;
   struct timerequest *timermsg;

   timerport-CreatePort(0,0);
   if( timerport -- NULL)
   {
      return(NULL);
   }

   timermsg-(struct timerequest *)
   CreateExtIO( timerport , sizeof( struct timerequest ) );
   if(timermsg -- NULL)
   {
      DeletePort(timerport);
      return(NULL);
   }

   error - OpenDevice( TIMERNAME,UNIT_VBLANK,timermsg,0);
   if(error !-0)
   {
      DeleteExtIO( timermsg );
      DeletePort( timerport);
      return(NULL);
   }
   return( timermsg );
}

WaitForTimer(tr,seconds,microseconds)
struct timerequest *tr;
ULONG seconds,microseconds;
{
   tr->tr_node.io_Command - TR_ADDREQUEST;
   tr->tr_time.tv_secs-seconds;
   tr->tr_time.tv_micro-microseconds;
   DoIO(tr);
   return(0);
}

DeleteTimer( tr )
struct timerequest *tr;
{
   struct MsgPort *tp;
   if( tr !- 0 )
   {
      tp - tr->tr_node.io_Message.mn_ReplyPort;
      if( tp!-0 )
      {
```

(continued)

ATTENTION **FUTURESOUND**™ OWNERS

MUSIC VISIONS©

Create a light show on your Amiga® Music Visions adds the visual dimension to music as it plays, using the FutureSound digitizer. Music Visions samples and analyzes the music, producing a light show in real-time!

Create a show to your taste by using a drawing program that creates IFF files called musicfields. Musicfields can be simple or complex; the only limit is your imagination! Options allow you to change the show as the music plays.

- *Bi-directional, variable speed, color cycling* • *3 display modes*
- *Uses any lo-res IFF picture* • *Adjustable color palette*
- *Variable frequency analysis* • *Requires FutureSound hardware*
- *Easy to use* • *Documentation included* • *and more!*

VISA/MasterCard accepted - no CODs please
Send $29.95 plus $3.00 shipping/handling to:
(Calif. residents add 6% sales tax)

DIGITAL WIZARDS, INC.

9307 Carlton Hills Blvd.
Santee, CA 92071
(619) 449-1281

©Music Visions, Copyright 1988, Digital Wizards, Inc.
FutureSound is a registered trademark of Applied Visions
® Amiga is a registered trademark of Commodore-Amiga, Inc.

AMIGA SOUND OASIS

The Amiga™ family of computers posess incredible sound capabilities that have been ignored due to a lack of quality samples. Now, using the Amiga's disk drive and the **Amiga Sound Oasis** software, you can read any of over 2500 professional sounds in the Mirage™ Sample Library. Play sounds live through the Amiga's stereo outputs using a MIDI keyboard or incorporate Mirage™ sounds into any music program that reads IFF samples, including Dynamic Studio and Dynamic Drums.

LIST PRICE $99.95 NOT COPY PROTECTED

Call or write:

NEW WAVE SOFTWARE
PO Box 438 St. Clair Shores, MI 48080
(313) 771-4465

Mirage is a trademark Ensoniq Inc.

```
         CloseDevice(tr);
         DeleteExtIO( tr,sizeof(struct timerequest) );
         DeletePort(tp);
      }
   }
   return(0);
}
```

Listing Three: Comp.asm

```
   CSECTtext
   XREF _myhandler
   XDEF _HandlerInterface

_HandlerInterface:
   MOVEM.L A0/A1,-(A7)
   JSR _myhandler
   ADDQ.L #8,A7
   RTS

   END
```

Listing Four: Event.c

```
#include <exec/types.h>
#include <exec/ports.h>
#include <exec/memory.h>
#include <exec/io.h>
#include <exec/tasks.h>
#include <exec/interrupts.h>
#include <devices/input.h>
#include <exec/devices.h>
#include <devices/inputevent.h>


#define F1KEYUP 0xD0
struct InputEvent copyevent; /* Local copy of event */
       /* assumes never has a next.event attached */

struct MsgPort *inputDevPort;
struct IOStdReq *inputRequestBlock;
struct Interrupt handlerStuff;

struct InputEvent dummyEvent;

extern struct MsgPort *CreatePort();
extern struct IOStdReq *CreateStdIO();

struct MemEntry me[10];

   /* If we want the input handler itself to add anything to the
   * input stream, we will have to keep track of any dynamically
   * allocated memory so that we can later return it to the system.
   * Other handlers can break any internal links the handler puts
   * in before it passes the input events.
   */

struct InputEvent
*myhandler(ev,mydata)
struct InputEvent *ev;    /* and a pointer to a list of events */
struct MemEntry *mydata[];  /* system will pass me a pointer to my
          * own data space.
          */
{
   /* demo version of program simply reports input events as
   * it sees them; passes them on unchanged. Also, if there
   * is a linked chain of input events, reports only the lead
   * one in the chain, for simplicity.
   */
   if(ev->ie_Class==IECLASS_TIMER)
   {
      return(ev);
   }
   /* don't try to print timer events!!! They come every 1/10th sec. */
   else
   {
      Forbid();
      copyevent.ie_Class = ev->ie_Class;
      copyevent.ie_SubClass = ev->ie_SubClass;
      copyevent.ie_Code = ev->ie_Code;
      copyevent.ie_Qualifier = ev->ie_Qualifier;
      copyevent.ie_X = ev->ie_X;
      copyevent.ie_Y = ev->ie_Y;
      copyevent.ie_TimeStamp.tv_secs = ev->ie_TimeStamp.tv_secs;
      copyevent.ie_TimeStamp.tv_micro = ev->ie_TimeStamp.tv_micro;
      Permit();
   }

   /* There will be lots of events coming through here;
   * rather than make the system slow down because something
   * is busy printing the previous event, lets just print what
   * we find is current, and if we miss a few, so be it.
   *
   * Normally this loop would "handle" the event or perhaps
   * add a new one to the stream. (At this level, the only
   * events you should really be adding are mouse, rawkey or timer,
   * because you are ahead of the intuition interpreter.)
   * No printing is done in this loop(lets main() do it) because
   * printf can't be done by anything less than a 'process'.
   */
   return(ev);
   /* pass on the pointer to the event( most handlers would
   * pass on a pointer to a changed or unchanged stream)
   * (we are simply reporting what is seen, not trying to
   * modify it in any way) */
}

/* NOTICE: THIS PROGRAM LINKS ITSELF INTO THE INPUT STREAM AHEAD OF
 * INTUITION. THEREFORE THE ONLY INPUT EVENTS THAT IT WILL SEE AT
 * ALL ARE TIMER, KEYBOARD, and GAMEPORT. AS NOTED IN THE PROGRAM,
 * THE TIMER EVENTS ARE IGNORED DELIBERATELY */



extern struct Task *FindTask();
struct Task *mytask;
LONG mysignal;
extern void HandlerInterface();

struct timerequest *mytimerRequest;
extern struct timerequest *PrepareTimer();
extern int WaitForTimer();
extern int DeleteTimer();

main()
{
   SHORT error;
   ULONG oldseconds,oldmicro,oldclass;

   /* init dummy event, this is what we will feed to other handlers
    * while this handler is active */

   dummyEvent.ie_Class = IECLASS_NULL; /* no event happened */
   dummyEvent.ie_NextEvent = NULL;     /* only this one in the chain */
   SetTaskPri( FindTask(0), 20);
   inputDevPort = CreatePort(0,0);     /* for input device */
   if(inputDevPort == NULL) exit(-1);  /* error during createport */
   inputRequestBlock = CreateStdIO(inputDevPort);
   if(inputDevPort == 0) { DeletePort(inputDevPort); exit(-2); }
            /* error during createstdio */

   mytimerRequest = PrepareTimer();
   if(mytimerRequest==NULL) exit(-3);

   handlerStuff.is_Data = (APTR)&me[0];
        /* address of its data area */
   handlerStuff.is_Code = HandlerInterface;
        /* address of entry point to handler */
   handlerStuff.is_Node.ln_Pri = 51;
        /* set the priority one step higher than
         * Intuition, so that our handler enters
         * the chain ahead of Intuition.
         */
   error = OpenDevice("input.device",0,inputRequestBlock,0);
   if(error==0) printf("\nOpened the input device");

   inputRequestBlock->io_Command = IND_ADDHANDLER;
   inputRequestBlock->io_Data = (APTR)&handlerStuff;

   DoIO(inputRequestBlock);
   copyevent.ie_TimeStamp.tv_secs = 0;
   copyevent.ie_TimeStamp.tv_micro = 0;
   copyevent.ie_Class = 0;
   oldseconds=0;
   oldmicro=0;
   oldclass=0;

   for(;;)   /*FOREVER*/
   {
      WaitForTimer(mytimerRequest,0,100000);
         /* TRUE=wait; time=1/10th second */

      if(copyevent.ie_Class==IECLASS_RAWKEY &
         copyevent.ie_Code==F1KEYUP) break;
      else
      {
         Forbid();
         if(copyevent.ie_TimeStamp.tv_secs != oldseconds ||
         copyevent.ie_TimeStamp.tv_micro != oldmicro ||
         copyevent.ie_Class != oldclass)
         {
            oldseconds = copyevent.ie_TimeStamp.tv_secs;
            oldmicro = copyevent.ie_TimeStamp.tv_micro;
            oldclass = copyevent.ie_Class;
            showEvents(&copyevent);
         }
         Permit();
      }
   }

   /* remove the handler from the chain */
   inputRequestBlock->io_Command = IND_REMHANDLER;
   inputRequestBlock->io_Data = (APTR)&handlerStuff;
   DoIO(inputRequestBlock);

   /* close the input device */
   CloseDevice(inputRequestBlock);

   /* delete the IO request */
   DeleteStdIO(inputRequestBlock);

   /* free other system stuff */
   DeletePort(inputDevPort);
   DeleteTimer(mytimerRequest);
           /* end of main */
}

int
showEvents(e)
struct InputEvent *e;
{
   printf("\n\nNew Input Event");
   printf("\nie_Class = %lx",e->ie_Class);
   printf("\nie_SubClass = %lx",e->ie_SubClass);
   printf("\nie_Code = %lx",e->ie_Code);
   printf("\nie_Qualifier = %lx",e->ie_Qualifier);
   printf("\nie_X = %ld",e->ie_X);
   printf("\nie_Y = %ld",e->ie_Y);
   printf("\nie_TimeStamp(seconds)=%lx",e->ie_TimeStamp.tv_secs);
   return(0);
}

struct timerequest *
PrepareTimer()
{
   int error;
   struct MsgPort *timerport;
   struct timerequest *timermsg;

   timerport=CreatePort(0,0);
   if( timerport == NULL)
   {
      return(NULL);
   }
```

Now Open In Texas!

The most complete Amiga store in the country-
with the most competitive pricing.

Over 700 software titles in stock.
Wide selection of hardware, accessories, and books.
Write or call for a catalog.

Come be AMAZED!

10am-6pm Saturday & Sunday April 23 & 24.
Door prizes and special events including demos and
talks by leading software developers.

Amazing Computer Systems, Inc.

Frankford Village Shopping Center
3030 North Josey Lane Suite 144
(2 doors south of Skaggs)
Carrollton, Texas 75007
(214) 394-8383

Open Mon.-Sat. 10am-6pm Thurs. 10am-8pm

Amazing Computer Systems is not affiliated in any way with PiM Publications, Inc.
or Amazing Computing.

```
   timermsg=(struct timerequest *)
   CreateExtIO( timerport , sizeof( struct timerequest ) );
   if(timermsg == NULL)
   {
      DeletePort(timerport);
      return(NULL);
   }

   error = OpenDevice( TIMERNAME,UNIT_VBLANK,timermsg,0);
   if(error !=0)
   {
      DeleteExtIO( timermsg );
      DeletePort( timerport);
      return(NULL);
   }
   return( timermsg );
}

WaitForTimer(tr,seconds,microseconds)
struct timerequest *tr;
ULONG seconds,microseconds;
{
   tr->tr_node.io_Command = TR_ADDREQUEST;
   tr->tr_time.tv_secs=seconds;
   tr->tr_time.tv_micro=microseconds;
   DoIO(tr);
   return(0);
}

DeleteTimer( tr )
struct timerequest *tr;
{
   struct MsgPort *tp;
   if( tr != 0 )
   {
      tp = tr->tr_node.io_Message.mn_ReplyPort;
      if( tp!=0 )
      {
         CloseDevice(tr);
         DeleteExtIO( tr,sizeof(struct timerequest) );
         DeletePort(tp);
      }
   }
   return(0);
}
```

•AC•

Modula-2 Programming *on the Amiga™*

Termination modules for the Benchmark and TDI compilers

by Steve Faiwiszewski

In a previous article I discussed the M2Amiga package's termination code feature and its merits. I would like to present termination modules for the Benchmark and TDI compilers, and discuss their implementation.

Modula-2 programmers are quite familiar with the initialization code section of implementation modules. This code is guaranteed to run upon startup, before the main module gets a chance to execute. The initialization section enables the programmer to set up things specific to the implementation module without having to do so explicitly from the main module. This feature can make software a great deal more modular and "clean." Unfortunately, there is no analogous mechanism for program shutdown, which forces the main module to "know" about any cleanup that must be done by the other modules, and to call those modules' cleanup procedures explicitly.

The M2Amiga package introduced the ability to specify for each module a procedure to be called upon program exit. This frees the main (or any other "client") module from having to know about any cleanup details internal to the implementation module, allowing the programmer to create truly reusable, stand-alone code. M2Amiga's implementation of this mechanism is quite elegant; upon the program's termination—whether in the normal course of the program or due to some runtime error—all the implementation modules' designated termination procedures are executed.

This facility is especially useful on the Amiga, where any resource allocated by a task must be explicitly deallocated by it, or be lost forever. (Or at least until the next reboot.) If your program is compiled with an Amiga compiler other than M2Amiga, and it encounters a runtime error and dies, all its open screens and windows will be left open, and its allocated memory will be lost.

Following M2Amiga's example, I set out to implement this termination feature for the Benchmark and the TDI compiler. Specifying the terminating procedures is quite simple, as Modula-2 has procedure variables. Executing is another story. I could not find a way to have these routines implicitly execute upon normal program exit. Therefore, a program needing to use this feature must not exit at the bottom of its code or call HALT; it must call the procedure ExitGracefully. This is the only thing in my implementation that differs from the way M2Amiga does it (after all, theirs is integrated at the language implementation level while mine is done as user code).

To call the exit procedure upon encountering runtime errors, you must be able to trap these errors. The TDI package provides a very simple way to do so; Benchmark's approach is a bit more difficult. In the TDI package, there is a procedure variable in module AMIGAX called ErrorProcessor which is called when runtime errors occur. To trap these errors, all I had to do was point ErrorProcessor to one of my own procedures. Since my goal was to call my own cleanup routines PLUS the usual code of Trapper and the Post Mortem Debugger, I simply saved the value of ErrorProcessor in another procedure variable first, then pointed ErrorProcessor to my own code. When an error occurs and the cleanup code runs, I simply call ErrorProcessor's old value (by actually calling the other procedure variable).

Since ExitGracefully must be the last procedure to run, it must not return (meaning that execution cannot just fall out the bottom of the procedure). Modula-2 provides the keyword HALT to terminate execution but unfortunately, the TDI package treats a HALT as a runtime error, so it cannot be used. Instead, the AmigaDOS Exit procedure is used. Exit returns control to the CLI, setting up the process's exit status according to the value passed to Exit as an argument.

Since the runtime package does not provide for it, trapping errors with the Benchmark package requires a bit more work. Each task on the Amiga has a field in its Task record, called tcTrapCode, that points to an error handler. It's quite simple to point tcTrapCode to my own code. But since I still wanted to execute the error handling code of the Benchmark RuntimeErrors module, something more complicated was required.

When an exception occurs, the task's trap frame and the trap number are pushed onto the stack, and the error handler code is executed. (The exception I'm referring to is the Motorola definition of a runtime error. In Amiga lingo, a runtime error invokes a trap; an exception is something different.) The error handler must at least pop the trap number of the stack before executing an RTE (Return from Exception). Since I wanted to execute Benchmark's own error handler, I had to make sure the processor's stack was set up correctly before the Benchmark error handler executed. For one thing, my code can't simply call the error handler, as the compiler will generate a JSR (jump to subroutine) instruction which will alter the stack by pushing the return address onto it. Instead, I had to code an actual jump to the error handler using the INLINE statement.

The compiler always generates some instructions at the beginning of a procedure to push certain things on the stack, so I had to generate code to negate that. I found when absolute addressing is used, all the compiler generates is a LINK instruction (therefore, it generated more code when base-relative addressing— the default— was used). Then all I had to do was code an UNLK instruction. LINK allocates a temporary area on the stack (called the stack frame), and UNLK deallocates it.

In the Benchmark package, a HALT is a perfectly acceptable event, so ExitGracefully uses it to exit the program. The status code is returned to the CLI by simply setting up the CLIReturnCode variable imported from module System. The AmigaDOS Exit procedure should NOT be called. It will bypass Benchmark's own exit code, where libraries are closed and other cleanup is done.

Lastly, it's important to note, the Benchmark compiler defaults to having all runtime error checking disabled. I strongly urge you to always run with error checking enabled—it can save hours of detective work. Error checking can be enabled using compiler switches, or it can be permanently enabled using the M2Init utility.

Listing One

```
DEFINITION MODULE BenchmarkTermination;
        (* * * * * * * * * * * * * * * * * * * * * * * * * *)
        (* A Module to handle runtime error as well as     *)
        (* to provide means for easy graceful termination  *)
        (* for the Benchmark Modula-2 compiler.            *)
        (* * * * * * * * * * * * * * * * * * * * * * * * * *)
(*$L+*) (* Very IMPORTANT!!  *)
         (* Must use absolute addressing for this to work! *)


PROCEDURE ExitGracefully(status : LONGINT);
(* Call all the modules' designated 'terminator' *)
(* procedures, and then terminate the program,   *)
(* returning status to the CLI.                  *)


PROCEDURE AddTerminator(p : PROC);
(* Add procedure p to the list of designated     *)
(* terminator procedures which are called upon   *)
(* program exit.                                 *)


PROCEDURE CtrlC(): BOOLEAN;
(* See if a Ctrl-C signal has been detected.     *)

END BenchmarkTermination.
```

Listing Two

```
IMPLEMENTATION MODULE BenchmarkTermination;
        (* * * * * * * * * * * * * * * * * * * * * * * * * *)
        (* A Module to handle runtime error as well as     *)
        (* to provide means for easy graceful termination  *)
        (* for the Benchmark Modula-2 compiler.            *)
        (* * * * * * * * * * * * * * * * * * * * * * * * * *)

(*$L+*) (* Very IMPORTANT!!  *)
         (* Must use absolute addressing for this to work! *)

FROM AmigaDOS  IMPORT SigBreakCtrlC;
FROM Tasks     IMPORT TaskPtr, FindTask, CurrentTask;
FROM System    IMPORT CLIReturnCode;
FROM SYSTEM    IMPORT ADDRESS, ADR, INLINE, SETREG;
FROM RunTimeErrors
            IMPORT InstallErrorHandler, RemoveErrorHandler;

CONST
     JmpIndirectA0 - 4ED0H; (* jmp (A0) instruction *)
     UnlinkA5 - 4E5DH;               (* UNLK A5 instruction  *)
     A0 - 8;

VAR
    RuntimeErrorOccurred : BOOLEAN;
    Myself               : TaskPtr;
    OldTrapHandler       : ADDRESS;
    TerminatorIndex      : CARDINAL;
    Terminators          : ARRAY[1..50] OF PROC;

PROCEDURE TheEnd;
BEGIN
    Myself^.tcTrapCode :- OldTrapHandler;
    IF NOT RuntimeErrorOccurred THEN
          RemoveErrorHandler;
          HALT
    END;
END TheEnd;

PROCEDURE CtrlC(): BOOLEAN;
(* see if control-c signal has arrived *)
BEGIN
    RETURN SigBreakCtrlC IN Myself^.tcSigRecvd
END CtrlC;


PROCEDURE ExitGracefully(status : LONGINT);
VAR
    i : CARDINAL;
BEGIN
    CLIReturnCode :- status;
    FOR i :- TerminatorIndex TO 1 BY -1 DO
          Terminators[i]
    END;
END ExitGracefully;

PROCEDURE MyErrorHandler;
(* A runtime error was encountered.        *)
(* Cleanup and call the old error handler *)
BEGIN
    RuntimeErrorOccurred :- TRUE;
    ExitGracefully(9999);     (* Set return code to 9999 *)
    INLINE(UnlinkA5);  (* Undo the compiler's LINK instruction *)

(* now jump to the old trap handler *)

    SETREG(A0,OldTrapHandler);
    INLINE(JmpIndirectA0)
END MyErrorHandler;

PROCEDURE AddTerminator(p : PROC);
BEGIN
    INC(TerminatorIndex);
    Terminators[TerminatorIndex] :- p;
END AddTerminator;

BEGIN
    TerminatorIndex :- 0;
    AddTerminator(TheEnd);
    InstallErrorHandler;
    Myself :- FindTask(0);
(* Save the old error handler, and then point *)
(* the trap vector to our error handler       *)
    WITH Myself^ DO
          OldTrapHandler :- tcTrapCode;
          tcTrapCode :- ADR(MyErrorHandler);
    END;
    RuntimeErrorOccurred :- FALSE;
END BenchmarkTermination.
```

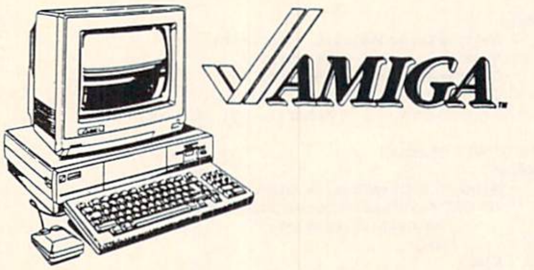
Listing Three

```
DEFINITION MODULE TdiTermination;
        (* * * * * * * * * * * * * * * * * * * * * * * * * *)
        (* A Module to handle runtime error as well as     *)
        (* to provide means for easy graceful termination  *)
        (* for the TDI Modula-2 compiler.                  *)
        (* * * * * * * * * * * * * * * * * * * * * * * * * *)

PROCEDURE CtrlC(): BOOLEAN;
(* See if a Ctrl-C signal has been detected.     *)

PROCEDURE ExitGracefully(status : LONGCARD);
(* Call all the modules' designated 'terminator' *)
(* procedures, and then terminate the program,   *)
```

(continued)

Century Systems

has the best prices and service in the USA on the incredible AMIGA Computer.



Many Amiga's and Amiga expansion products available.

Prices too low to print. And service too good to ignore.

Century Systems
8033 University Ave.
Des Moines, IA 50311

SALES 1-800-223-8088

```
(* returning status to the CLI.                       *)

PROCEDURE AddTerminator(p : PROC);
(* Add procedure p to the list of designated         *)
(* terminator procedures which are called upon       *)
(* program exit.                                      *)

END TdiTermination.
```

Listing Four

```
IMPLEMENTATION MODULE TdiTermination;
       (* * * * * * * * * * * * * * * * * * * * * * * * *)
       (* A Module to handle runtime error as well as     *)
       (* to provide means for easy graceful termination  *)
       (* for the TDI Modula-2 compiler.                  *)
       (* * * * * * * * * * * * * * * * * * * * * * * * *)

FROM InOut              IMPORT WriteString, WriteLn;
FROM LongInOut          IMPORT WriteLongHex;
FROM DOSLibrary         IMPORT SIGBreakC;
FROM DOSExtensions      IMPORT Process;
FROM DOSProcessHandler IMPORT Exit;
FROM Tasks              IMPORT TaskPtr, FindTask;
FROM AMIGAX             IMPORT ErrorProcessor, ProcessPtr;
IMPORT Trapper;

TYPE
    ProcPtr = POINTER TO Process;

VAR
    RuntimeErrorOccurred : BOOLEAN;
    Myself                : ProcPtr;
    OldErrorProcessor     : PROC;
    TerminatorIndex       : CARDINAL;
    Terminators           : ARRAY[1..50] OF PROC;
    ExitStatus            : LONGCARD;

PROCEDURE TheEnd;
BEGIN
    ErrorProcessor := OldErrorProcessor;
    IF NOT RuntimeErrorOccurred THEN
        Exit(ExitStatus)
    END;
END TheEnd;

PROCEDURE CtrlC(): BOOLEAN;
(* see if control-c signal has arrived *)
BEGIN
    RETURN SIGBreakC IN Myself^.prTask.tcSigRecvd
END CtrlC;

PROCEDURE ExitGracefully(status : LONGCARD);
VAR
    i : CARDINAL;
BEGIN
    ExitStatus := status;
    FOR i := TerminatorIndex TO 1 BY -1 DO
        Terminators[i]
    END;
END ExitGracefully;

PROCEDURE MyErrorHandler;
(* A runtime error was encountered.          *)
(* Cleanup and call the old error handler *)
BEGIN
    RuntimeErrorOccurred := TRUE;
    ExitGracefully(9999);        (* Set return code to 9999 *)
    OldErrorProcessor;
END MyErrorHandler;

PROCEDURE AddTerminator(p : PROC);
BEGIN
    INC(TerminatorIndex);
    Terminators[TerminatorIndex] := p;
END AddTerminator;

BEGIN
    TerminatorIndex := 0;
    AddTerminator(TheEnd);
    Myself := ProcessPtr;
    OldErrorProcessor := ErrorProcessor;
    ErrorProcessor := MyErrorHandler;
    RuntimeErrorOccurred := FALSE;
END TdiTermination.
```

Listing Five

```
MODULE BmTest;

(* A little program to test the termination and      *)
(* error handling module for the Benchmark compiler *)

FROM BenchmarkTermination IMPORT AddTerminator, ExitGracefully;
FROM TermInOut             IMPORT WriteString, WriteLn;

VAR
    a,b,c : CARDINAL;

PROCEDURE bar;
BEGIN
    WriteString('This is the End'); WriteLn;
END bar;

BEGIN
    AddTerminator(bar);
    a := 1;
    b := 2;
    c := 0;
    a := b DIV c;
END BmTest.
```

The following listings were accidently ommited from AC v3.3. We apologize for the inconvenience. - Ed.

```
DEFINITION MODULE EtchIntuiStuff;
(*********************************************************)
(*      Intuition-related routines for EtchAsketch       *)
(*                                                       *)
(*        Written for the Benchmark M2 compiler.         *)
(*                                                       *)
(* Steve Faiwiszewski                    December 1987   *)
(*********************************************************)

FROM Tasks IMPORT SignalSet;

PROCEDURE SetSpriteColors(num : INTEGER);
(* Set the color registers for hardware sprite 'num' *)

PROCEDURE ListenToIntuition(VAR MySignalSet : SignalSet);
(* Set up the SignalSet for listening to Intuition   *)
(* messages which are received through the window's *)
(* userport.                                         *)

PROCEDURE ProcessIntuiMessages(VAR exiting : BOOLEAN);
```

```
(* Process any intuition messages that might have    *)
(* been received.                                    *)

END EtchIntuiStuff.



IMPLEMENTATION MODULE EtchIntuiStuff;
(*********************************************************)
(*      Intuition-related routines for EtchAsketch      *)
(*                                                       *)
(*        Written for the Benchmark M2 compiler.         *)
(*                                                       *)
(* Steve Faiwiszewski                      December 1987 *)
(*********************************************************)


FROM EtchGlobal
                IMPORT AddTerminationProc, ExitGracefully,
                       MyVPort, MyRPort, MyScreen;
FROM Views      IMPORT ColorMap, ColorMapPtr, ViewModesSet,
                       Hires, Sprites, SetRGB4;
FROM Intuition IMPORT Screen, ScreenPtr, NewScreen,
                       CustomScreen, WindowPtr, OpenWindow,
                       CloseWindow, NewWindow,
                       WindowFlags, WindowFlagsSet,
                       GadgetFlags, GadgetFlagsSet,
                       IDCMPFlags, IDCMPFlagsSet,
                       IntuiMessagePtr,
                       IntuiText, IntuiTextPtr,
                       MenuNull, Menu, MenuPtr,
                       MenuFlags, MenuFlagsSet, MenuItem,
                       MenuItemPtr,
                       MenuItemFlags, MenuItemFlagsSet,
                       MenuItemMutualExcludeSet, HighComp,
                       SetMenuStrip, ClearMenuStrip,
                       MENUNUM, ITEMNUM,
                       OpenScreen, CloseScreen,
                       ShowTitle, ScreenToBack;
FROM SYSTEM     IMPORT ADR, ADDRESS, BYTE, TSIZE;
FROM Tasks      IMPORT SignalSet;
FROM TermInOut IMPORT WriteString, WriteLn;
FROM Ports      IMPORT GetMsg, MessagePtr, ReplyMsg;
FROM Rasters    IMPORT DrawModeSet, Jam2, RastPortPtr;
FROM Drawing    IMPORT SetAPen, RectFill;

CONST
    WIDTH = 320;
    HEIGHT = 200;
    DEPTH = 3;
    MENUWIDTH = 110;

VAR
    newScr      : NewScreen;
    MyWindow    : WindowPtr;
    MenuStrip   : MenuPtr;
    MyMenu      : Menu;
    MyMenuItems : ARRAY[0..3] OF MenuItem;
    MyMenuText  : ARRAY[0..3] OF IntuiText;
    CurrentColor: CARDINAL;


PROCEDURE CleanUp(line : ARRAY OF CHAR; n : CARDINAL);
BEGIN
    IF n > 2 THEN
        ClearMenuStrip(MyWindow^);
        CloseWindow(MyWindow^)
    END;
    IF n > 1 THEN CloseScreen(MyScreen^) END;
    WriteString(line); WriteLn;
END CleanUp;

PROCEDURE InitMenuRec (VAR Amenu : Menu;
                          left, top, width, height : INTEGER;
                          text : ADDRESS) : MenuPtr;
(* Initialize a menu record.          *)
BEGIN
    WITH Amenu DO
      NextMenu := NIL;
      LeftEdge := left;   TopEdge := top;
      Width := width;     Height := height;
      Flags := MenuFlagsSet{MenuEnabled};
      MenuName := text;
      FirstItem := NIL
    END;
    RETURN (ADR(Amenu))
END InitMenuRec;

PROCEDURE InitItemRec (VAR menitem : MenuItem;
                          left, top, width, height : INTEGER;
                          Cmd : CHAR;
                          ItemFillPtr : ADDRESS): MenuItemPtr;
(* Set up a menu item record. *)
BEGIN
    WITH menitem DO
        NextItem := NIL;
```

(continued)

HARD DRIVES

PIONEER has the storage solution for your personal computer, business system or graphics workstation. Your requirements are met by our product family. From the 20-megabyte through the high performance 150-megabyte drives, PIONEER spans the storage spectrum to offer the right capacity, performance and price to suit your system needs.

All PIONEER hard drive have an embedded SCSI controller build-in. This makes you compatible with other SCSI systems on the market (no interfaces needed between the computer host controller and your hard drive).

SCSI HOST CONTROLLER

The SCSI host/controller has been designed with growth and expansion in mind. The hardware and software provided with your SCSI host/controller will allow you to connect, set-up and access information from any SCSI device. The SCSI host/controller allows AmigaDOS to deal with your hard drive at a system level using the same commands that you are now using with the standard Amiga 3.5" floppy drive.

The SCSI host/controller will provide you with the fastest, most reliable and fully featured hard drive system available for the AMIGA 500, 1000 and 2000. The SCSI host/controller comes complete with a expansion bus pass-thru, 18 inch cable, 50-pin SCSI bus and Apple/Mac Pseudo SCSI 25-pin connectors.

HARD DRIVE CASE

Full-height case comes complete with a 45 watt fan cooled power supply, switchable 110/220 AC voltage (50/60 Hz), room for one full-height or two half-height drives, power-on lED, half-height cover plate and power cord. Dimensions: 7" wide, 5" high and 15.5" deep.

Half-height case comes complete with a 45 watt fan cooled power supply, switchable 110/220 AC voltage (50/60 Hz), room for one half-height drive or tape back-up, convenient carry handle and power cord. Dimensions: 6" wide, 3.25" high and 15.5" deep.

OUR GUARANTEE TO YOU:
All our products are 100% guaranteed. We will replace, exchange or refund*
any purchase to your satisfaction.
* Refund must be made within 15 days of receipt of product.

Hard Drive comes complete with drive, SCSI host/controller, your choice of case and a ONE YEAR WARRANTY!

20 meg	$ 659.95
40 meg	$ 879.95
65 meg	$ 949.95

External 3.5" Floppy Drive

PIONEER'S 3.5" floppy drive comes complete with an 18" cable, fully enclosed metal case, a pass-thru for addional drives, single or dual drive, ninety day warranty and consumes about 2.1 watts.

Single Drive	$ 159.95
Dual Drive	$ 329.95

PIONEER COMPUTING

-Blazing a New Frontier for AMIGA COMPUTERS-

2469 East 7000 South #200
Salt Lake City, UT 84121
(801) 942-1174

D - Five Associates
19 Crosby Drive
Bedford, MA
01730-0523
(617) 275-8892
Tired of the high cost of computer repairs?
→ FLAT Labor charges
→ FREE Estimates
→ Warranty work
Also:
1764 to 512K: $61.95
128 64K vdc RAM: $40.00
NEW: C=1902 conversion to RGB-I: $40.00
$19.95*
C-64/128 & peripherals
29.95*
Amiga 1000
$34.95*
Commodore PC-10
* plus parts and sales tax
Authorized Commodore Service Center

```
        LeftEdge := left;
        TopEdge := top;
        Width := width;
        Height := height;
        Flags := MenuItemFlagsSet{ItemText, ItemEnabled} +
                          HighComp;
        MutualExclude :=  MenuItemMutualExcludeSet{};
        ItemFill := ItemFillPtr;
        SelectFill := NIL;
        Command := BYTE(Cmd);
        IF Cmd <> 0C THEN
            Flags := Flags + MenuItemFlagsSet{CommSeq}
        END;
        SubItem := NIL;
        NextSelect := 0;
    END;
    RETURN ADR(menitem)
END InitItemRec;

PROCEDURE InitTextRec (VAR it : IntuiText;
                       left, top : INTEGER;
                       front, back : BYTE;
                       Mode : DrawModeSet;
                       text : ADDRESS) : IntuiTextPtr;
(* Initialize a menu text record. *)
BEGIN
    WITH it DO
        FrontPen := front;
        BackPen := back;
        LeftEdge := left;
        TopEdge := top;
        DrawMode := Mode;
        ITextFont := NIL;
        IText := text;
        NextText := NIL
    END;
    RETURN ADR(it);
END InitTextRec;

PROCEDURE SetSpriteColors(num : INTEGER);
(* Set up the color regs for the chosen simple sprite *)
VAR
    x : CARDINAL;
BEGIN
    x := (num DIV 2) * 4;
    SetRGB4 (MyVPort^, 17+x, 14, 3, 0);   (* red        *)
    SetRGB4 (MyVPort^, 18+x, 15, 9, 7);   (* peach      *)
    SetRGB4 (MyVPort^, 19+x, 5, 13, 13);  (* light blue *)
END SetSpriteColors;

PROCEDURE SetColors (sp : ScreenPtr);
VAR
    i : CARDINAL;
BEGIN
    WITH sp^ DO
        SetRGB4(ViewPort, 0, 0, 0, 0);     (* black      *)
        SetRGB4(ViewPort, 1, 5, 13, 13);   (* light blue *)
        SetRGB4(ViewPort, 2, 6, 5, 10);    (* purple     *)
        SetRGB4(ViewPort, 3, 14, 3, 0);    (* red        *)
        SetRGB4(ViewPort, 4, 13, 11, 8);   (* tan        *)
        SetRGB4(ViewPort, 5, 5, 13, 0);    (* green      *)
        SetRGB4(ViewPort, 6, 15, 9, 7);    (* peach      *)
        SetRGB4(ViewPort, 7, 15, 15,15);   (* white      *)
    END
END SetColors;

PROCEDURE OpenMyScreen(W,H,D : CARDINAL);
BEGIN
    WITH newScr DO
        LeftEdge := 0;
        TopEdge := 0;
        Width := W ;
        Height := H;
        Depth := D;
        DetailPen := BYTE(0);
        BlockPen := BYTE(1);
        IF W > 320 THEN
            ViewModes := ViewModesSet{Hires};
        ELSE
            ViewModes := ViewModesSet{};
        END;
        ViewModes := ViewModes + ViewModesSet{Sprites};
        Font := NIL;
        DefaultTitle := NIL;
        Gadgets := NIL;
        CustomBitMap := NIL;
    END;
    newScr.Type := CustomScreen;
    MyScreen := ScreenPtr(OpenScreen(newScr));
    IF MyScreen = NIL THEN
        CleanUp('Could not open screen!',1);
        ExitGracefully
    END;
    MyVPort := ADR(MyScreen^.ViewPort);
    ShowTitle(MyScreen^,FALSE);
    SetColors(MyScreen);
END OpenMyScreen;

PROCEDURE InitMenus(VAR MenuStrip : MenuPtr);
(* Create the menu *)
BEGIN
    MenuStrip := InitMenuRec(MyMenu, 3, 0, 78, 10,
                             ADR(" Action"));

    MyMenu.FirstItem :=
      InitItemRec(MyMenuItems[0],0,0,MENUWIDTH,10,'B',
        InitTextRec(MyMenuText[0],0,1,BYTE(0),BYTE(1),Jam2,
                        ADR("Push Back")));

    MyMenuItems[0].NextItem :=
      InitItemRec(MyMenuItems[1],0,10,MENUWIDTH,10,'N',
        InitTextRec(MyMenuText[1],0,1,BYTE(0),BYTE(1),Jam2,
                        ADR("Next Color")));

    MyMenuItems[1].NextItem :=
      InitItemRec(MyMenuItems[2],0,20,MENUWIDTH,10,'E',
        InitTextRec(MyMenuText[2],0,1,BYTE(0),BYTE(1),Jam2,
                        ADR("Erase")));

    MyMenuItems[2].NextItem :=
      InitItemRec(MyMenuItems[3],0,30,MENUWIDTH,10,'Q',
        InitTextRec(MyMenuText[3],0,1,BYTE(0),BYTE(1),Jam2,
                        ADR("Quit")));
END InitMenus;

PROCEDURE OpenMyWindow(W,H,D : CARDINAL);
VAR
    MyNewWindow : NewWindow;
BEGIN
    WITH MyNewWindow DO
        LeftEdge := 0;
        TopEdge := 0;
        Height := H;
        Width := W;
        DetailPen := BYTE (0);
        BlockPen := BYTE (1);
        Title := NIL;
        Flags :=
            WindowFlagsSet{Activate,Borderless,BackDrop};
        IDCMPFlags := IDCMPFlagsSet{MenuPick};
        Type := CustomScreen;
        CheckMark := NIL;
        FirstGadget := NIL;
        Screen := MyScreen;
        BitMap := NIL;
        MinWidth := 0; MinHeight := 0;
        MaxWidth := 0; MaxHeight := 0;
    END;
    (* Now open the window *)
    MyWindow := OpenWindow(MyNewWindow);
    IF MyWindow = NIL THEN
        CleanUp('Could not open window!',2);
        ExitGracefully
    END;
    InitMenus(MenuStrip);
    SetMenuStrip(MyWindow^,MenuStrip^);
    MyRPort := MyWindow^.RPort;
END OpenMyWindow;

PROCEDURE ListenToIntuition(VAR MySignalSet : SignalSet);
BEGIN
    INCL(MySignalSet,
        CARDINAL(MyWindow^.UserPort^.mpSigBit))
END ListenToIntuition;

PROCEDURE SelectNextColor;
BEGIN
```

```
      INC(CurrentColor);
      IF CurrentColor > 7 THEN CurrentColor := 1 END;
      SetAPen(MyRPort^,CurrentColor)
  END SelectNextColor;

  PROCEDURE EraseScreen;
  (* Erase the screen to black *)
  BEGIN
      SetAPen(MyRPort^,0);
      RectFill(MyRPort^,0,0,WIDTH-1,HEIGHT-1);
      SetAPen(MyRPort^,CurrentColor)
  END EraseScreen;

  PROCEDURE ProcessMenus(MenuNum : CARDINAL;
                         VAR Quit : BOOLEAN);
  VAR
      item : CARDINAL;
  BEGIN
      IF MENUNUM(MenuNum) = 0 THEN
          item := ITEMNUM(MenuNum);
          CASE item OF
              0 : ScreenToBack(MyScreen^)          |
              1 : SelectNextColor                  |
              2 : EraseScreen
              ELSE
                  Quit := TRUE
          END;
      END;
  END ProcessMenus;

  PROCEDURE ProcessIntuiMessages(VAR exiting : BOOLEAN);
  VAR
      mp      : MessagePtr;
      imp     : IntuiMessagePtr;
      class   : IDCMPFlagsSet;
      code    : CARDINAL;
  BEGIN
      mp := GetMsg(MyWindow^.UserPort^);
      WHILE mp <> NIL DO
          imp := IntuiMessagePtr(mp);
          class := imp^.Class;
          code := imp^.Code;
          ReplyMsg(mp);
          IF MenuPick IN class THEN
              IF code <> MenuNull THEN
                  ProcessMenus(code,exiting);
              END;
          END;
          mp := GetMsg(MyWindow^.UserPort^);
      END;
  END ProcessIntuiMessages;

  PROCEDURE CloseWindowAndScreen;
  BEGIN
      CleanUp('Releasing Screen/Window Resources',3)
  END CloseWindowAndScreen;

  BEGIN
      CurrentColor := 1;
      OpenMyScreen(WIDTH,HEIGHT,DEPTH);
      OpenMyWindow(WIDTH,HEIGHT,DEPTH);
      AddTerminationProc(CloseWindowAndScreen);
  END EtchIntuiStuff.
  MODULE EtchAsketch;


  (**********************************************************)
  (*                    EtchAsketch                          *)
  (*   A program to demonstrate the use of the joystick      *)
  (*            device and hardware sprites.                 *)
  (*       Written for the Benchmark M2 compiler.            *)
  (* Steve Faiwiszewski                     December 1987   *)
  (**********************************************************)

  FROM EtchGlobal    IMPORT AddTerminationProc, ExitGracefully,
                            MyVPort, MyRPort, ChipAllocate;
  FROM EtchJoystick IMPORT OpenJoystick, GetJoystickStatus,
                            ListenToJoystick,  Left, Right,
                            PrepareToReadJoystick,
                            SetTriggerTime, Forward, Backward;
  FROM EtchIntuiStuff
                    IMPORT ListenToIntuition, SetSpriteColors,
                            ProcessIntuiMessages;
  FROM Tasks        IMPORT SignalSet, Wait;
  FROM Drawing      IMPORT Move, Draw, SetAPen,PolyDraw, SetDrMd,
                            RectFill, ReadPixel;
  FROM Sprites      IMPORT SimpleSprite, GetSprite, ChangeSprite,
                            FreeSprite, MoveSprite, AnySprite;
  FROM Gels         IMPORT InitGels, GelsInfoPtr, VSprite,
                            VSpritePtr;
  FROM Rasters      IMPORT RastPortPtr;
  FROM TermInOut    IMPORT WriteString, WriteLn;
  FROM SYSTEM       IMPORT TSIZE;

  CONST
     SpriteHeight = 5;
     SpriteCenterOffset = (SpriteHeight DIV 2);
```

APL.68000

$99

A HIGHLY OPTIMIZED ASSEMBLER BASED APL INTERPRETER FOR FAST AND POWERFUL PROGRAMS. FEATURES A COMPLETE INTERFACE TO THE AMIGA ENVIRONMENT WITH PULL-DOWN MENUS, REQUESTER AND ALERT BOXES, SPEECH, SOUND AND GRAPHICS FACILITIES.

Order direct for $99 + $7 shipping, $10 Canada. VISA/MC/AMEX + 4% NJ res + 6% sales tax.

SPENCER
ORGANIZATION, INC.
P.O. Box 248 Westwood, N.J. 07675
(201) 666-6011

```
     Xinc = 1;
     Yinc = 1;
     Xmin = SpriteCenterOffset;
     Ymin = 5;
     Xmax = 320 - Xmin;
     Ymax = 190 - Ymin;

  TYPE
     DirectionType = (up,down,left,right);
     DirectionSet  = SET OF DirectionType;
     SpriteImageBuf = RECORD
         data : ARRAY[0..SpriteHeight+1],[0..1] OF CARDINAL;
     END;

  VAR
     SpriteImagePtr : POINTER TO SpriteImageBuf;
     MySignalSet    : SignalSet;
     MySprite       : SimpleSprite;
     CurX, CurY,
     StolenSprite   : INTEGER;

  PROCEDURE GetDirection(VAR NewDirection : DirectionSet;
                         VAR ButtonDown : BOOLEAN);
  (* Get Joystick directions and button press *)
  VAR
     Ystick,
     Xstick : INTEGER;
  BEGIN
     GetJoystickStatus(ButtonDown,Xstick,Ystick);
     NewDirection := DirectionSet{};
     CASE Ystick OF
         Forward  : INCL(NewDirection,up)          |
         Backward : INCL(NewDirection,down)
         ELSE     (* do nothing *)
     END; (* case *)
     CASE Xstick OF
         Left  : INCL(NewDirection,left)           |
         Right : INCL(NewDirection,right)
         ELSE     (* do nothing *)
     END; (* case *)
  END GetDirection;

  PROCEDURE Allowed(NewDirection : DirectionType) : BOOLEAN;
  (* Check if requested movement is allowed *)
  BEGIN
     CASE NewDirection OF
        up    : RETURN CurY >= Ymin + Yinc             |
        down  : RETURN CurY <= Ymax - Yinc             |
        left  : RETURN CurX >= Xmin + Xinc             |
        right: RETURN CurX <= Xmax - Xinc
        ELSE
              RETURN FALSE
     END; (* case *)
  END Allowed;

  PROCEDURE MovePlayer(NewDirection : DirectionSet;
                       ButtonDown : BOOLEAN);
  VAR
     dir : DirectionType;
  BEGIN
     IF NewDirection <> DirectionSet{} THEN
         FOR dir := up TO right DO
             IF (dir IN NewDirection) AND Allowed(dir) THEN
                 CASE dir OF
                    up    : DEC(CurY,Yinc)  |
```

(continued)

GOMF!

'...it is small, installs with just a double-click on an icon...a welcome utility for Amiga users.'

-INFO magazine

'...Without a doubt, the most important program this month is GOMF. All Amiga users are sick and tired of losing running programs and data to the dreaded Guru, and anything that can stop this terrible syndrome deserves a round of applause!'

'...So if Gurus are getting to you, add GOMF to your Amiga environment and you can start saying goodbye to the GURU too!'

-Commodore magazine

Now GOMF is even better!
Version 2.0 NOW shipping

NEW FEATURES include:
- Automatic removal of ALL errors (task held, GURU, etc.)
- Preserves low memory auto vectors.
- NUKE those pesky CANCEL requestors once and for ALL!
- Eliminate ANY task or device AT ANY TIME.
- auto-configures for 68000/68010/68020.
- CLI and Workbench compatible.
- Extensive, easy to understand documentation.

only $39.95

New Products for All *AMIGAS*
from **HyperTek/Silicon Springs**

Deluxe MIDI interface

Simply the BEST full-featured MIDI interface available for the AMIGA 500/1000. Fully compatible with all programs that use serial MIDI standard output, ONLY the DELUXE MIDI INTERFACE offers you: SERIAL PORT Pass-thru, MIDI IN, MIDI THRU, and two SWITCHABLE THRU/OUTS ! This is the optimum configuration for anyone with more than one synth or other MIDI device. No THRU on your synth(s)? No problem! Simply flip a switch on the Deluxe MIDI interface, and avoid costly THRU boxes! Includes 6 foot cable, 1 year warranty.

Please Specify model 500/1000........$89

TTL Hi-res Monitor Adapter

For the Amiga 500/1000/2000. Plugs into RGBI port for ULTRACRISP flicker-free high resolution monochrome video output. Perfect for HI-RES GRAPHICS, WORDPROCESSING, CAD, DESKTOP PUBLISHING, BUSINESS, etc. ANY application in ANY resolution is SHARPER and CLEARER with the TTL Hi-res Monitor Adapter. Perfect for use with the low-cost (under $100) Commodore 1901 monitor or Monitor 80. Includes easy installation instructions, a disk with a special Hi-Res WorkBench font and RGB port pass-thru (a 1080 RGB monitor may be used simultaneously). 1 year warranty.

Be KIND to your EYES!

(Monitor not included)........$99.95

Light Pen

Designed to work with ANY Amiga program, the LIGHT PEN and DRIVER allows use of both your mouse and a pro-quality LIGHT PEN for the ultimate in precision graphics. Perfect for PAINTING, DRAWING, freehand SKETCHING, CAD and virtually ALL other AMIGA programs. Software features include: Single-pixel precision, variable sensitivity, ZOOM mode, button toggle, etc. Includes Inkwell Systems' DT-184A high quality two button lightpen. *MAIL ORDER SPECIAL* While supplies last, we'll also include a handy PEN-HOOK that attaches easily to your monitor to hold the AMIGA LIGHTPEN when not in use!
(500/1000/2000)........$129.95

All products are NOW SHIPPING! Please make cheque or money order payable to:

HyperTek/Silicon Springs
Suite 205-2571 Shaughnessy St. Port Coquitlam, BC CANADA V3C 3G3
Phone (604) 942-4577 Dealer Inquires invited.

All orders add $4 shipping. All prices in US$
Order by phone! VISA, Mastercard, AMEX welcome.
U.S. orders add $3 handling fee.

AMIGA is a registered trademark of Commodore-Amiga, Inc.

```
                    down : INC(CurY,Yinc)   |
                    left : DEC(CurX,Xinc)   |
                    right: INC(CurX,Xinc)
                END;
            END; (* if *)
        END; (* for *)
        IF ButtonDown THEN
            Draw(MyRPort^,CurX + SpriteCenterOffset,
                          CurY + SpriteCenterOffset);
        ELSE
            Move(MyRPort^,CurX + SpriteCenterOffset,
                          CurY + SpriteCenterOffset);
        END;
        MoveSprite(MyVPort^,MySprite,CurX,CurY);
    END;
END MovePlayer;

PROCEDURE LoopAround;
VAR
    sig : SignalSet;
    Directions : DirectionSet;
    exiting,
    ButtonDown : BOOLEAN;
BEGIN
    exiting := FALSE;
    ButtonDown := FALSE;
    REPEAT
        PrepareToReadJoystick;
        sig := Wait(MySignalSet);
        ProcessIntuiMessages(exiting);
        GetDirection(Directions,ButtonDown);
        MovePlayer(Directions,ButtonDown)
    UNTIL exiting;
END LoopAround;

PROCEDURE InitSpriteImage;
VAR
    i : CARDINAL;
BEGIN
    ChipAllocate(SpriteImagePtr,TSIZE(SpriteImageBuf));
    WITH SpriteImagePtr^ DO
        data[1,0] := 2000H;
        data[2,0] := 7000H;       data[2,1] := 2000H;
        data[3,0] := 0D800H;       data[3,1] := 5000H;
        data[4,0] := 7000H;       data[4,1] := 2000H;
        data[5,0] := 2000H;
    END; (* with *)
END InitSpriteImage;

PROCEDURE PrepareSprite;
BEGIN
    InitSpriteImage;
    StolenSprite := GetSprite(MySprite,AnySprite);
    IF StolenSprite = -1 THEN
        WriteString('*** Could not obtain sprite! ***');
        WriteLn;
        ExitGracefully
    END;
    SetSpriteColors(StolenSprite);
    CurX := (Xmax - Xmin) DIV 2 + Xmin;
    CurY := (Ymax - Ymin) DIV 2 + Ymin;
    WITH MySprite DO
        height := SpriteHeight;
        x := CurX;
        y := CurY;
    END;
    ChangeSprite(MyVPort^,MySprite,SpriteImagePtr);
    Move(MyRPort^,CurX + SpriteCenterOffset,
                  CurY + SpriteCenterOffset);
END PrepareSprite;

PROCEDURE cleanup;
BEGIN
    FreeSprite(StolenSprite)
END cleanup;

BEGIN
    PrepareSprite;
    AddTerminationProc(cleanup);
    OpenJoystick;
    MySignalSet := SignalSet{};
    ListenToIntuition(MySignalSet);
    ListenToJoystick(MySignalSet);
    IF NOT SetTriggerTime(1) THEN
        WriteString('*** error setting trigger ***');
        WriteLn;
        ExitGracefully
    END;
    LoopAround;
    ExitGracefully
END EtchAsketch.
```

•AC•

• BOOK REVIEWS •

by Richard Grace

Learning C: Programming Graphics On the Amiga and Atari ST

By Marc Sugiyama & Christopher Metcalfe
Published by Compute! Books, $18.95

C Primer Plus, Revised Edition

By Mitchell Waite, Stephen Prata, & Donald Martin
Published by Howard W. Sams, $24.95

Inside The Amiga With C

By John Thomas Berry
Published by Howard W. Sams, $22.95

One of the books about programming in the Amiga environment is *Learning C: Programming Graphics On the Amiga and Atari ST*. Marc Sugiyama and Chris Metcalfe have put together a volume introducing reader to the C language in general, and specifically to C programming on the Amiga and the Atari ST. The book includes many large programs which must be typed in to create a custom programming environment compatible with the book's concepts.

This approach has its problems. While the Amiga and Atari ST are both 68000-based machines, they are very different in their overall hardware structure. As a result, the actual programming tools available to programmers on the respective machines are different. The ST does not have the complex set of custom graphics chips (blitter, copper, etc.) or the extensive ROM kernel the Amiga does. The ST also does not have an equivalent of a CLI or even of the Intuition operating system. The Amiga's Exec task management kernel, which is the core of the multi-tasking system, also has no Atari ST counterpart.

As a result, by trying to promote common ground between the two machines, this book completely ignores all these Amiga qualities. The closest the authors come to acknowledging this discrepancy is in the introduction: "The Amiga has a very complex set of screen-controller chips which fill areas, draw lines, and move images around the screen, giving the Amiga a speed advantage over the ST. The Amiga does in hardware what the ST must do in software; leaving the Amiga's processor more time to do computing." This observation is accurate, but doesn't solve the problem.

Rather than presenting a useful synopsis and introduction to the Amiga Intuition and ROM Kernel environments (written in C), the authors substitute several graphics programs to fulfill the functions of line and polygon drawing, fills, windowing, etc. "in an effort to bring these two machines together." This approach is fine for the Atari, but the Amiga programmer is left out in the cold with no idea how many aspects of his powerful machine can be used. Keystroking in a graphics routine is redundant for Amiga owners, when far more powerful manipulations can be found inside the machine..

This flaw is fatal, at least from the Amiga standpoint. Otherwise, the book is fairly well written. The aspects mentioned above are essential to understanding the Amiga. This book would certainly be more meaningful for the ST programmer, because Sugiyama and Metcalfe do a fairly good job of describing, in readable form, some of the often cryptic ins and outs of the C language.

The expertise the authors bring to this work is evident. The concept is flawed, however. It is hard to understand why the authors decided on their approach, given the differences between the machines. Amiga programmers interested in C would do well to stay away from this book because they will learn nothing about what really makes their machine tick.

There are superior alternatives, however. One is the *C Primer Plus* by Mitchell Waite, Stephen Prata, and Donald Martin. While it is in no way Amiga-specific, this book provides an excellent introduction and overview of C. It is exceptionally readable, given the notorious tendency of C specific books to be as cryptic as their subject matter. In fact, it is tempting to call this text the best of its kind. Any budding Amiga C programmer will profit handsomely from this book.

(continued)

Once familiar with the C language, the Amiga programmer is confronted with the onerous task of navigating the 2000 pages of Addison-Wesley's famous texts to understand the complex Amiga system. No longer. Those books are still very useful, but John Thomas Berry's *Inside The Amiga With C* provides, for the first time, a useful, well-written, one-volume programmer's "roadmap" of the Amiga.

Berry aptly summarizes: "The multiprocessing capability of the Amiga puts it in a class beyond most small computers. One important thing that experienced programmers miss when making the translation from mainframe or minicomputers to small systems is the ability to run more than one task at a time. From a productivity standpoint, the advantage is obvious."

Berry does not attempt to teach C programming; it is not his task. An exhaustive exposition of the ROM kernel is also beyond the scope of his work. However, he discusses many key ROM functions, gives a detailed account of Intuition (in one chapter!), and gives the reader the signposts and information necessary to acquire a comprehensive knowledge of the Amiga's numerous programming tools. The chapter on Intuition alone covers almost a hundred pages and follows this logic: "Of the three programming environments found on the Amiga—Intuition, AmigaDOS and the executive kernel—this is the most difficult one in which to develop software. It is a very complex system which requires even small, simple programs to handle many display details. However, Intuition is meant to be the primary user operating system, and most software will have to deal with it."

> Amiga programmers, especially in C code, face a challenging task when they begin to explore the Amiga environment.

The book is very well written and approachable by any intelligent user. However, some prior knowledge of C is necessary (the more the better). For the Amiga C programmer this text is really the next logical step after *C Primer Plus* because it puts that earlier knowledge into the proper context.

Amiga programmers, especially in C code, face a challenging task when they begin to explore the Amiga environment. Fortunately The Waite Group, through Howard W. Sams publishing, has provided us with the means to make that task a little easier.

•AC•

AMIGA 500 COMMAND CENTER

NEW

OUTSTANDING FEATURES:

- ► ORGANIZES WORK SPACE
- ► CONVENIENT LOCATION OF DRIVES & MONITOR
- ► HANDY DISK STORAGE
- ► MUCH MORE, FOR ONLY 99.95

DELUXE MODEL FEATURES:

- ► BUILT-IN POWER STRIP
- ► SURGE PROTECTION & NOISE FILTER
- ► BUILT-IN COOLING FAN
- ► FINGERTIP CONTROL OF COMPUTER & PERIPHERALS
- ► MUCH MORE, FOR ONLY 179.95

COMMAND CENTER ALSO AVAILABLE FOR:

128	124.95
64	109.95
64C	109.95

ACCESSORIES:

PHOENIX HEAVY DUTY POWER SUPPLY	
SUPRA 20 MB DRIVE	
SUPRA 30 MB DRIVE	
JOY STICK EXTENSION	12.50
TILT SWIVEL MONITOR STAND	14.95
DUST COVER	19.95
ELECTRICAL UPGRADE KIT	89.95

ASK ABOUT OUR *NEW* ACOUSTICAL PRINTER ENCLOSURE — THE "SOUNDTRAP"

ONE OF THE TOP COMMODORE ACCESSORIES NOW AVAILABLE FOR AMIGA 500!

* AMIGA is a trademark of Commodore Business Machines. Computer not included.

KETEK

P.O. Box 203, Oakdale, IA 52319
FOR FAST SERVICE, CALL:
1-800-626-4582
1-319-338-7123 (IN IOWA)

SHIPPING AND HANDLING:

Continental U.S.	APO/FPO/Canada/Puerto Rico/Hawaii/Alaska
$4.50 for each Command Center	$13.00 for each Command Center
$2.00 for one accessory item	$ 4.00 for one accessory item
$3.00 for two or more accessories	$ 6.00 for two or more accessories

68000 Assembly Language for the Amiga

by Chris Martin

Last month I presented a program that set up and displayed a simple, non-Intuition screen. However, much remains to be explained in our discussion of "display routines" on the Amiga Kernel.

Variations in Display

You will recall from last month's discussion, ViewPorts may have special display properties of their own. Each ViewPort may have its own dimensions, color set, and display mode (resolution). Therefore, a display consisting of more than one ViewPort could have variations in display modes. For example, one half of the screen may be interlaced and hi-res with a set of 16 colors, while the other half may be low-res with only two colors. To accomplish this, the program must "link" the ViewPorts together so the display routines can process the entire display. The element of the ViewPort structure titled "vp_Next" is a pointer to a linking ViewPort. The order in which ViewPorts must be linked is related to the position of the ViewPort on the screen. For example, picture a two ViewPort display: one ViewPort on the top portion of the screen and one on the bottom. The ViewPort on the top must be the first in the linked list and must "point to" the second ViewPort (in the lower portion of the screen). Remember, the computer constructs the display from top to bottom. A list of ViewPorts must also be constructed this way. Each ViewPort may have its own display mode. The display modes, defined in the ViewPort element "vp_Modes," are as follows:

V_HIRES	Set high resolution graphics mode (640 pixels across).
V_LACE	Set interlace mode (400 pixels down).
V_HAM	Set Hold-and-Modify mode (4096 colors at once).
V_SPRITES	Include sprites (animation graphics objects) on screen.
V_DUALPF	Set dual playfield mode (discussed later).
V_PFBA	Set alternate priority of playfields (discussed later).

Modes can be combined by placing a "!" between the modes. For example, to create a display with graphics of 640 x 400 pixels, the modes V_HIRES and V_LACE must be combined, and the following definition is used:

V_HIRES ! V_LACEc

Since most of the work is done by the Amiga's display routines, creating variations in the display is rather simple. However, one must remember the order in which the display routines should be called. I spent many hours trying to determine why some of the programs I wrote didn't work, and I found the fault in the way I set up the display. Here is the correct order:

1. Allocate memory space for bit-planes, use AllocRaster().

2. Initialize the View structure with InitView().

3. Point the View structure to the first ViewPort in the linked list. Set the View element v_ViewPort to the location of the ViewPort structure.

4. Initialize the ViewPort structure(s) with InitVPort().

5. Link the ViewPort list if there is more than one ViewPort in the View.

6. Point each ViewPort to its own RasInfo structure. Set the ViewPort element vp_RasInfo to the location of the RasInfo structure.

7. Set width, height, and display mode of each ViewPort.

8. Initialize the BitMap structure with InitBitMap().

9. Point the RasInfo structure to the BitMap structure by setting the RasInfo element ri_BitMap to the location of the BitMap.

10. Make the ViewPort by calling MakeVPort(). You need to "Make" only the first ViewPort in the linked list, since the others will be pointed to and included in the "Make" process.

(continued)

11. Modify the Copper instruction set by merging the new Copper display list with the currently existing list. Use the MrgCop() instruction.

12. Show the display with the LoadView() instruction.

Smooth Scrolling

Smooth scrolling, a graphic effect used in many games to achieve a smooth moving background, is created by allowing the screen display to act as a "window" to a larger graphic area. The large display is shown only a portion at a time, the smooth scrolling effect is attained by moving this "window" to display other parts of the large display. What I call the "large display" is a bitmap of dimensions larger than the ViewPort, the "window" is the ViewPort itself, which can display only a portion of the bitmap (see figure). Each ViewPort has its own RasInfo structure, pointed to by the vp_RasInfo element of the ViewPort. The RasInfo elements determine the position of the "window" in the large bitmap: ri_RxOffset and ri_RyOffset are the X and Y coordinates of the upper left corner of the ViewPort in relation to the upper left corner of the bitmap. The smooth scrolling effect is attained by incrementing or decrementing the ri_RxOffset and ri_RyOffset values to move the position of the ViewPort in the large bitmap. After the RxOffset and RyOffset elements have been modified, the display must be recreated to give the effect of motion. The following routines must be called in order to re-make the display:

```
MakeVPort()
MrgCop()
LoadView()
```

Dual-Playfields

Dual-playfield mode is a very interesting feature of the Amiga's graphics capabilities. In dual-playfield mode, you have two superimposed screens (playfields) to work with. Each playfield can have a maximum of 8 colors. You must define the following structures for EACH playfield: RasInfo, BitMap, and RastPort (for drawing routines).

Perhaps the most important structure in the definition of a dual-playfield display is the RasInfo structure. For a dual-playfield display, you must create two RasInfo structures — one for each playfield. The RasInfo structures must be linked together in the same manner that ViewPorts are linked: the element ri_Next of the first RasInfo structure should be set as a pointer to the second RasInfo structure. As in a single playfield display, each RasInfo structure has its own BitMap. (An interesting effect in dual-playfield mode is the smooth-scrolling of each playfield at the same time. Two BitMap structures also must be defined, each pointed to by the RasInfo structures. Just as in a single playfield display, the BitMaps may be of dimensions up to 1024 x 1024 pixels. When drawing in each playfield, you must address a RastPort structure for each. I simply name the RastPorts "rp1" and "rp2," for the first and second playfields' RastPorts, respectively.

Because of the many ways of distributing bit-planes among each playfield, there are seven different ways to configure a dual-playfield display. The following table shows the assignment of bit-planes in dual-playfield mode:

# of Bit-planes	Playfield 1 Depth	Playfield 2 Depth
0	0	0
1	1	0
2	1	1
3	2	1
4	2	2
5	3	2
6	3	3

Another consideration you must make is the display priority of each playfield; in other words, which playfield will appear "on top of" the other. By default, the first playfield (playfield 1) will have priority over the second. However, this priority may be switched by setting the V_PFBA flag in the ViewPort Modes. In any case, you must set the V_DUALPF flag in the ViewPort Modes to tell the system that this ViewPort will use dual-playfields. Follow these steps to create a dual-playfield display:

1. Set up a View structure.

2. Allocate memory for two BitMaps.

3. Set up two linked RasInfo structures, each pointing to its own BitMap.

4. Set up one ViewPort structure (both playfields will share the same ViewPort).

5. Set the ViewPort element vp_RasInfo to point to the first RasInfo structure in the linked list.

6. Initialize each BitMap with InitBitMap().

7. Initialize the ViewPort with InitVPort().

8. Set the V_DUALPF and, if you want, the V_PFBA flags in the ViewPort Modes.

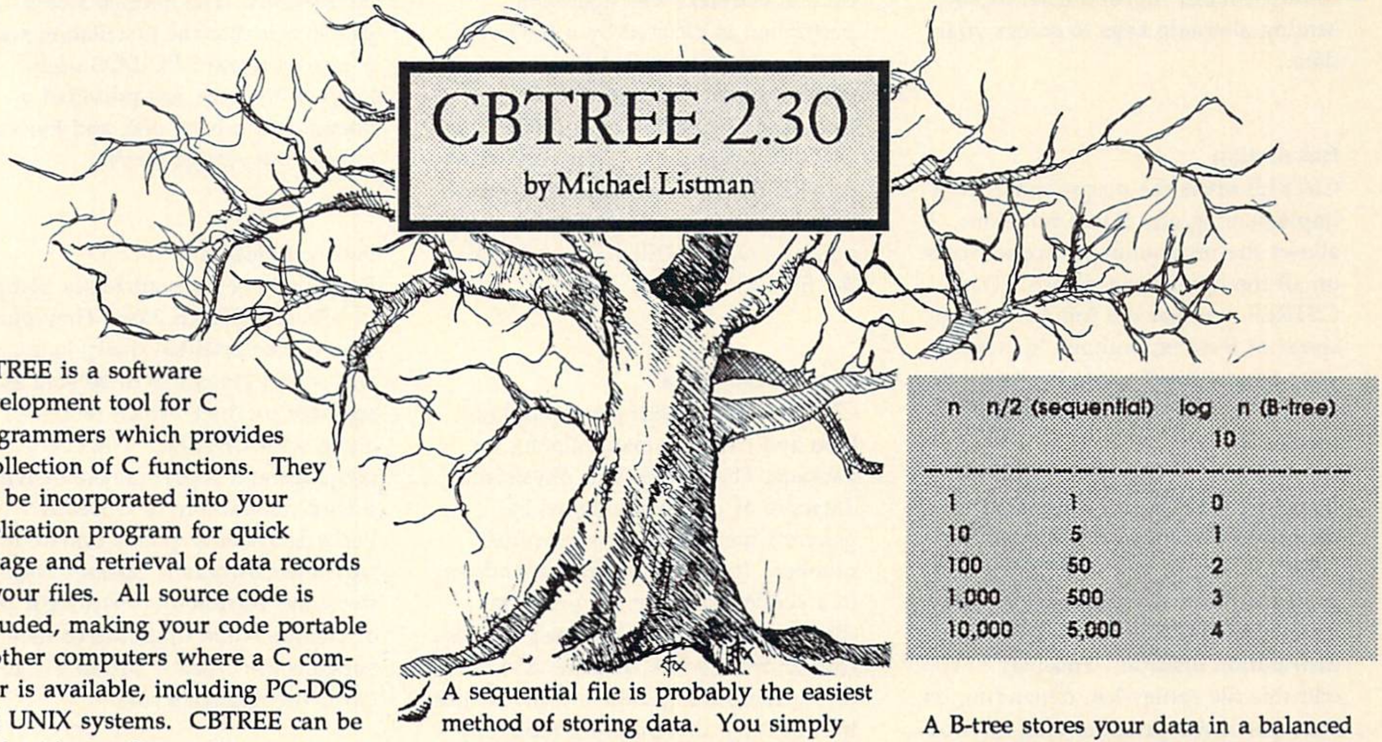
9. Call MakeVPort(), MrgCop(), and LoadView().

Next month, I will present a program that sets up a dual-playfield display and smooth scrolls both playfields at once!

•AC•

CBTREE 2.30

by Michael Listman



CBTREE is a software development tool for C programmers which provides a collection of C functions. They can be incorporated into your application program for quick storage and retrieval of data records in your files. All source code is included, making your code portable to other computers where a C compiler is available, including PC-DOS and UNIX systems. CBTREE can be used with any C compiler now available for the Amiga.

What You Get

The CBTREE package includes an interface to B+tree manipulation routines, sample programs which use the CBTREE routines to manipulate a database, a number of utility programs for creating and customizing your B+tree file structures, a user's manual, and free telephone support from Peacock Systems. Also included is a command file to install CBTREE on your system. All the documentation is provided on a single disk.

Background

CBTREE provides an interface to routines which manipulate a B+tree. Before describing the tools in this package, I will discuss some of the properties of B-trees and B+trees, and the pros and cons of using such an access method. Three operations are normally performed on data records stored in a file. These are: FIND a record (given its key), INSERT a new record, and DELETE an existing record.

A sequential file is probably the easiest method of storing data. You simply place each record consecutively in a file. FINDing a record using a sequential file is easy, but slow, unless there's a small number of records. To find a particular record, we scan the file from beginning to end. We examine about half the records in the file. Given n records, we perform approximately n/2 operations. INSERTing records in a sequential file is simple. Position yourself at the end of the file and write the new record. This is a single operation. However, to avoid duplicates, we first have to do a FIND to make sure the record does not exist, and then insert it. DELETE can be implemented by FINDing the record and then marking it deleted (a single bit is sufficient for this)—again roughly n/2 operations.

A B-tree is a data record indexing system which provides fast access to your data. Each of the operations in the elementary operation can be implemented in a B-tree, and performed with order log n disk accesses. Refer to the table for the estimated number of file accesses required to retrieve data using a simple sequential file. Compare it to the number of file accesses using a B-tree index.

n	n/2 (sequential)	$\log_{10}$ n (B-tree)
1	1	0
10	5	1
100	50	2
1,000	500	3
10,000	5,000	4

A B-tree stores your data in a balanced tree structure. Each node of the tree contains pointers to its children, as well as the key of the smallest element descendant from the second child. The maximum number of children in a node is limited by the block size of the tree's nodes. All data records are stored at the leaves of the tree. (For more details on the internals of B-trees, see the description of 2-3 trees in Chapter 5 of *Data Structures and Algorithms*, by Aho, Hopcroft and Ullman, Addison-Wesley, 1983.) Storing your data records, using B-trees or any other indexing scheme, does use more disk space than a sequential file, but the time saved by the indexing should be well worth the extra space.

A B+tree (yes, that's "B plus tree") is a variation of a B-tree with the index and the data are stored in separate disk files. The B+tree leaves (index file) contain actual pointers to the record location in the data file, instead of containing the actual data themselves, as in a B-tree. This organization allows for any number of separate

(continued)

indices into your data files, each in a different index file, and each representing alternate keys to access your data.

Installation

CBTREE saves the programmer from implementing any B-tree functions. It allows the programmer to concentrate on all the application details. With CBTREE, you get the full power and speed of B+trees, without "getting your hands dirty" while implementing the low-level B+tree manipulation routines.

Installing CBTREE requires two formatted diskettes: one for the CBTREE library, and one for the program files. A command file for installing CBTREE is included on the distribution diskette. You may have to edit this file somewhat, depending on your particular programming environment, and also on which compiler you have. But the changes are quite straightforward.

Programming

Before creating a database, you must set up the CBTREE parameter file btparms.btr. A utility called btsetup is provided as a programmer interface for creating and updating this file. CBTREE allows you to change the parameters in btparms.btr using btsetup without recompiling your code. With this interface, you must provide the following information for each B+tree you are developing:

- *Name of the index file*
- *Size of each index block*
- *Maximum key length*
- *Data record size*

Given the information in the parameter file, your program must call the routine btrinit() to initialize each of the B+trees it will be using. It must also open all data and index files. At this point, all further calls are made to the routine cbtree(). The operation performed is dictated by a field you set in a structure called btcommo. The user's manual provides detailed examples of function calls for all CBTREE operations. Your code must provide the user interface and screen handling facilities for input and output, but CBTREE will handle all the file I/O.

Sample Programs

CBTREE comes with a sample database and code for manipulating the package. The sample is a physician's database of patients, indexed by patient's name and Social Security number. It stores both B+tree indices in a single index file. A program (Bld2keys) is supplied to read the data records from a raw data file and create the corresponding data and index files. In execution, this program took only about 45 seconds to build the database in RAM for 150 data records, but over 13 minutes for files on disk. Given the slow disk I/O of AmigaDOS, such a difference is expected.

The CBTREE package includes a program for manipulating this sample database. The user is asked which operation he would like to perform, as well as its keys and data, if required. The program then runs and displays the results. This program helps introduce CBTREE operations. Also included in the CBTREE package are programs for verifying the status of your B+trees and for timing your CBTREE operations.

User's Manual

The manual provided with version 2.30 of CBTREE is labelled version 2.00. Differences between this release and the manual are outlined in a separate document included in the package. The manual is 75 pages of explanation and examples of CBTREE function calls, as well as installation instructions. The manual's only problem is that the installation section is geared toward PC-DOS users. Peacock Systems has provided a document on both disk and hard copy to help out Amiga users.

Future Releases

Peacock Systems is currently planning to release CBTREE 3.00. They plan to offer a user-oriented query language and report generator to be sold as a separate product, which will work along with CBTREE. The query language and report generator will be priced comparably to CBTREE. Peacock Systems' policy update for registered owners is releases within the same version are distributed free of charge, while upgrades to new versions are charged $10 to $12 to cover the manual's cost.

Summary

CBTREE is an excellent product and a quality tool for the C programmer. Getting familiar with the CBTREE function calls takes some time. But the manual provides detailed examples that fully outline what must be set up before each call is made, and how to make each function call. Overall, CBTREE is not hard to use and the time spent getting familiar with the product is well spent.

•AC•

Peacock Systems, Inc.
2108-C Gallows Road
Vienna, Virginia 22810
(703) 356-7029
(703) 847-1743

ROOMERS

by The Bandito

The National Computer Graphics Association (NCGA) show was held recently in Anaheim, California, right next to Disneyland. (For anybody interested in computer graphics, the NCGA show was better.) Most of the products on display at the show were incredibly expensive—so much so that when people saw what the Amiga could do, they had a hard time believing the price. Some people were actually checking the wires, like in the Wang commercial.

Commodore had a booth filled with third-party developers displaying some of the best graphics products for the Amiga—VideoScape, Sculpt and Animate 3D, Deluxe Productions, TV Text and TV Show, Pro Video Plus, Professional Page, Digi-View, and X-CAD. The only new item displayed was Digi-View 3.0 with overscan (and some other features promised as well). Digi-View is a software upgrade that should be available by the time you read this.

Professional Page was attracting attention from *Publish!*, a magazine devoted to desktop publishing that usually concentrates on IBM and Macintosh products. A crew from *Entertainment Tonight* was filming various demos for a special report on desktop video (air time unknown as yet). A sign of the times: Apple employees at Commodore's booth, inspecting Amiga products with a close eye. Sure, you can do some of that stuff with a Mac II (some day), but at 3 or 4 times the price. There's a Mac II buyer born every minute, though.

No one expected any surprises at the Hanover Fair, the big computer show in Germany. But Irving Gould, CEO of Commodore, provided one—he announced the Amiga 2500 and 3000, and even showed A2000 cases with those nameplates on them. The A2500 has a 68020 and the A3000 has a 68030. Throw in the new Fat Agnus chip and the revised Denise, and that's basically what they were. No word on price or availability, but the Bandito would expect the A2500 for the fall. Oh, and while Commodore watchers were surprised by the announcement, Commodore employees were even more surprised. Seems that Gould hadn't mentioned to the Amiga product managers he would be making these announcements. In fact, while Gould was telling the world about these new Amigas, Commodore employees were telling exhibitors at the NCGA show that no announcements would be made about new Amigas. You just don't know who to believe, these days.

In the second quarter (probably in June), Commodore will introduce the A2024 monitor, a 1008 x 800 monochrome display for current Amigas. The new monitor pulls some tricks with the standard Amiga chips to get its high resolution, dividing the screen up into four virtual areas. It will be great for desktop publishing, but you'll want to use large fonts—the screen is 13 inches diagonally and 100 dpi, so standard Amiga fonts are tiny. However, don't try any animation—it would look pretty strange on this monitor.

New Amigas will add another resolution mode—640 x 400 non-interlaced. Don't expect more colors until summer 1989, at least. When the next round of custom chips appear, the color resolution will probably be 15 bits per pixel, which the Bandito's slide rule says is just a hair over 32,000 colors. You'll get 256 colors out of this palette at once, and there may be a special HAM mode that allows all 32,000 colors on the screen at once. Oh, and new operating systems? Well, expect 1.3 this summer, and then 1.4 by year-end. Then again, you may be running UNIX on your Amiga—another nifty gimmick shown at the Hanover Fair.

The word on the wires is that some Amiga developers are secretly looking into "porting" their products to Macintosh and MS-DOS computers. Why? Well, there's almost 2 million Macs out there (200,000 Mac II's) and perhaps 15 million MS-DOS machines, so the potential market is much bigger than the current Amiga market. The Bandito hears two theories from Amiga developers to explain their interest in Macintosh and MS-DOS. One is that if you have a successful product on the Amiga, it's easier and more profitable to translate this sure hit to a bigger marketplace than it is to develop a new product. The other theory is that if the product didn't succeed in the Amiga marketplace, maybe it will do better when there are 4 or 30 times as many computers to sell it to (if you think this second theory sounds a bit funny, you're right).

Companies are working on IBM and Mac products under both theories. A word to the wise: the Amiga third-party market is full of garage entrepreneurs who started out by putting their product in a ziploc bag, and though

(continued)

35MM SLIDES FROM YOUR ARTWORK!

AMIGA
Specialists

TRU-IMAGE™

Professional 35mm Slides

◈ Now you can have reproduction and presentation quality slides of your work

◈ Distortion-free--fills in raster lines crisp bright colors, converts all IFF files

Now
Custom graphic art and illustration.

$10 each for your 1st to 4th slides.
5 to 9 slides-$8.50
Over 10 slides-$8.00
Add $2.00 for shipping.
New York residents add sales tax.
Call (212) 777-7609 FOR DETAILS
Ask for Ilene--or write TRU-IMAGE
P.O. Box 660, Cooper Station
New York, N.Y. 10276

Amiga is a trademark of Commodore-Amiga.

most Amiga developers have gone beyond that, they're still not in the weight class of Mac or IBM type companies. The companies in those markets tend to have a LOT of money and some very clever people. The big fish coming from the Amiga pond had better watch out for the great white sharks...

...and the sharks are already circling. Takeover rumors are hovering around several developers. Companies that have created successful products have gotten big enough (or have the technology) to attract some of the heavy hitters in the computer business, where acquisition has been all the rage for the past year or so. The Bandito hears that Apple has supplied some Mac II's to certain developers to get them to develop their applications for Apple (and possibly for other reasons?). Dollars are beckoning, and it's a good bet that some companies will grab while the grabbin' is good.

You can tell the software business is doing well when companies are hiring each other's employees. Electronic Arts, long notorious for taking the best employees away from other companies (usually Activision), is now seeing the cream of its crop skimmed off. The Bandito hears that more than a half-dozen EA employees have left for greener pastures (read *higher salaries*) at Activision. Not only that, EA seems to be the place to hire new marketing vice-presidents—first NewTek, then Activision, and now Accolade have all acquired people from EA to become marketing VPs. Not to be outdone, EA has hired Aegis' marketing VP, John Skeel, as a product manager for video and music products. 'Round and 'round they go, where they stop, nobody knows...

The Bandito hopes some of these Amiga developers have been working hard, because we haven't seen much in the way of new products lately. Delays are everywhere, and the RAM chip situation (availability tight, prices going through the roof) hasn't helped. Memory expansion board prices are going up, and new products that depend on RAM chips may be delayed or see price increases. The Bandito thinks the prices for the NewTek's Video Toaster, Mimetics' ReaSyn board, and Commodore's frame buffer board will all be higher than initially announced. Of course, in six months, the price of RAM chips should fall, and maybe the board makers will lower their prices (then again, maybe they won't). Hopefully, there will be a flood of products released at Comdex in May.

Whatever happened to Caligari? Perhaps you remember the demo (first shown at Siggraph in 1986), with the "camera" flying through an Amiga logo floating in space, then zooming into and around a set of three-dimensional shapes. The display was tremendously impressive, but the product has never appeared. So the Bandito tapped into the electronic grapevine to find out the true story. Seems that Electronic Arts picked up the rights some time ago, but Octree Software (the developer) has had problems keeping and finding programmers. The word is that the interface is good, but the rendering still leaves much to be desired (no ray-tracing). The development is taking so long that the product may well be obsolete by the time it gets to market. You snooze, you lose, as the saying goes.

Is there a curse on low-cost genlocks? First Commodore, now Mimetics has problems getting a reliable product at the low price point. The Bandito thinks someone should just put out a basic, reliable genlock at whatever it really costs, and not try to cut corners. Digital Creation's SuperGen works well, but it has some extra features that not everyone needs, and it costs $750. How about a basic genlock for $300? No frills, no muss, no fuss. The Bandito's got his money ready—any takers?

Commodore's new president, Max Toy, is trying hard to fix the problems he's inherited. He even has an assistant who asks questions of Commodore dealers, buyers, and developers in an effort to identify problem areas and suggest solutions. A laudable effort, since many of Commodore's problems can be solved just by listening to the people who actually use the products on a day-to-day basis. Let's do our part by trying to come up with constructive suggestions on how to make things work better. Meanwhile, the Bandito will do his part by continuing to talk about what's really happening in the Amiga market, so everybody has enough information to make intelligent suggestions. Besides, gossip is so much fun, isn't it?

•AC•

[The statements and projections presented in "Roomers" are rumors in the purest sense. The bits of information are gathered by a third party source from whispers inside the industry. At press time, they remain unconfirmed and are printed for entertainment value only. Accordingly, the staff and associates of Amazing Computing™ *cannot be held responsible for the reports made in this column.]*

In the Public Domain

by C.W. Flatte

Announcing Fred Fish disks 129 to 138!

Will it ever stop?! I certainly hope NOT! Announcing Fred Fish disks 129-138! Available NOW from your favorite PDS source. Let's jump right in!

Fred Fish 129

DosKwik version 2.0
(update to disk 103)
Tired of those long, slow loads from floppies? Then put the pedal to metal with DosKwik, by Gary Kemper. Comprised of the two programs ReadKwik and RiteKwik, DosKwik allows you to save and load files from floppies about 3 times faster than AmigaDOS. If that isn't enough, DosKwik allows you to save more data on a diskette than ever before – a whopping 901,080 bytes on one diskette. DosKwik also allows you to save massive files, that would normally only fit on a hard disk, on multiple diskettes (for that gigantic one and a half meg digitized sound file). You should note, however, that DosKwik does not store files in AmigaDOS format. DosKwik is available only in binary form, and is only accessible through the CLI.

MRBackup versions 2.0 & 2.1
(update to disk 128)
There's nothing worse than a hard drive crash... especially when there are no backups. MRBackup to the rescue. This hard disk backup utility by Mark Rinfrt is flexible and time-saving. It backs up individual directories, directory trees, even a whole disk easily. Incremental backups, based on last modified dates, are also suported. File compression is standard. It's no speed demon, but MRBackup is powerful and easy to use. Extensive documentation is included.
Version 2.0 includes the source.
Version 2.1 includes only the binary (the source is available from the author).

---

*The virus is still alive and kicking... Get rid of it for good with **VirusX**, on Fred Fish 137!*

Paintjet
Attention PaintJet owners! This one's for you! Finally, an "official" Hewlett Packard PaintJet printer driver, received directly from Hewlett Packard and placed in the public domain. This one is snag-free! It has been fully tested by H-P on the A500, A1000, and A2000. Great printing and it doesn't cost fifty bucks!

Patch versions 1.3 & 2.0
Here's a useful program. The official definition of patch, from the Contents file: "Two independent ports of the very useful Unix utility 'patch,' which applies context diffs to text files to automatically update them." Got it? Just as the definition states, there are two independent ports of Patch included on this disk. Patch version 1.3 was ported to the Amiga by Rick Coupland, and Patch version 2.0 was ported by Johan Widen. Both versions include the source.

Fred Fish 130

DirMaster version 1.1
(update to disk 108)
If you're like me, you have anywhere from twenty to fifty diskettes scattered on your desk. You have another two hundred or more in a shoebox— many of which aren't even labeled, which is another story. What is actually on those diskettes?

Catalog those disks with Greg Peters' Directory Master, the fastest, most comprehensive, and most versatile disk cataloguer available on the Amiga. Functions include Finding Duplicates, Display Filtering, Powerful Sorting and much more. Directory Master is fully mouse and menu-driven, and can be run from the WorkBench. This is a

(continued)

Video Visions

CUSTOM VIDEO
GRAPHIC ART
CONSTRUCTION KIT
FOR DESKTOP VIDEOS.

SCRIPTS - BACKDROPS - PARTS
ANIMATIONS

- AN ART DISK SERIES FOR YOUR AMIGA GRAPHIC PROGRAMS
- VOLUME ONE & TWO SHIPPING AS YOU READ THIS AD
- INQUIRE ABOUT FUTURE VOLUMES PREPARED CUSTOM-MADE FOR YOUR PARTICULAR GRAPHIC / VIDEO PROGRAM -2D OR 3D.

TWO DISKS PER VOLUME
$19.95 EACH FOR
VOLUME 1 OR 2,
PLUS $3.50 S&H,
Mass Residents Add
5% Sales Tax.

CV—DESIGNS
61 CLEWLEY ROAD
MEDFORD, MASS.
02155
617-396-8354

Amiga is a Trademark of Commodore-Amiga.
Video Visions is a Trademark of CV Designs.

superior product! To get you started, Directory Master also includes sample database files of the Amicus, MCA and BIX_28_Dec libraries. Now where is that printer driver I downloaded a few months ago? Gee, only two hundred disks to check. Arrhhhh!

Evo

Did you ever wish you could watch the evolution of man from 20 million years B.C. to the present? Evo shows the path of human evolution over 20 million years. In fact, a total of 601 crania can be displayed by clicking on a specific era on a time-line. Clicking on the "text" option displays a paragraph about the species currently displayed. Evo is a wonderful product by Steve Bonner. Well done! Evo includes the source in C.

By the way, did you know that man of 4 million years ago (Australopithecus afarensis) walked upright, was only about 4 feet tall, and had a brain of about 450cc (vs. 1400cc for modern man)?

Hp version 1.0

Calculate those complex numbers with the Hp RPN scientific calculator by Steve Bonner. This exceptional calculator is designed to look and function like a hand-held scientific model. Hp supports calculations in binary, octal, decimal, hexidecimal, floating point, and complex numbers. Hp also includes 32 registers for storing data and transcendental functions. Includes the source in C. Let's see...
1+1=2...2+2=4...

Mach version 1.6a
MachClk version 1.2a

Is your mouse just dragging along? Need a little speed? Then turbocharge your mouse with Mach, the mouse accelerator. Available only through the CLI, Mach boosts the mouse by a variable acceleration factor (0-9). Mach also gives your Amiga mouse the capabilities of a Sun mouse—the window the pointer is over becomes the active window. Also features the ability to program HotKeys (assign definitions to the function keys). Other features include screen blanking, Click-to-Front capability, and more. The disk also includes MachClk, with all the features of Mach plus a title bar clock, beeper, and on-line meter. The title bar clock displays the available memory and the time. The beeper, when activated, will beep at set intervals (from 0 to 60 minutes). The on-line meter keeps track of the time on-line and converts it into terms we can deal with much better than minutes— dollars (from your pocket). The rate is variable and can be easily set.

Mach/MachClk is definitely a don't miss! Another superior product by Polyglot software. (Includes the source to Mach 1.6a in C.)

PatEdit

Designing fill patterns for use with the Amiga SetAfPt macro call? Eliminate graph paper and design patterns up to 32 bits high with PatEdit, by Don Hyde. PatEdit is easy to use and makes full use of the Intuition interface. PatEd also supports the saving and loading of pattern files. The save function also creates a C header file which contains the code for the data which is required to pass to SetAfPt, SetAPen, and SetBPen. PatEdit includes the source in C.

QMan

Mandelbrot programs create beautiful graphics, but are usually painstakingly slow. "Quick Mandelbrot" or Qman, by Steve Bonner, is a speedy mandelbrot-type program which is written partially in 68000 assembler code to speed things up. QMan is runable from the Workbench and makes full use of the Intuition interface. This program is great for people new to mandelbrots (it won't discourage them with slow results).

Wow! That's neat!

Well, that about covers the first two disks, and I'm out of time. However, one final note before I close:

The virus is still alive and kicking... let's get rid of it for good! There are a few CLI-based virus checkers available in the public domain, which do a great job in ridding the virus from your diskettes. But, what if even the mere thought of having to use the CLI is even worse to you than having the virus on ALL your disks?

The solution? VirusX, by Steve Tibbett, which is available on Fred Fish 137. VirusX runs in the background and automatically checks all inserted disks for a nonstandard boot sector. These disks can optionally have their boot sector rewritten to remove the virus. More on this next month...

Until then...
Gotcha!

—C.W. Flatte

These Companies and 15,000 Amiga Users Joined AmiEXPO, The Amiga Event in New York and Los Angeles

A-Squared
A.X. Productions
Abacus Software
Accolade
Activision
Aegis Development
Amazing Computing
Amic Development Corp.
Amiga Science and Technology Users
Amiga Sentry
AmigaWorld
AmiNET, Inc.
AMuse, Inc.
Anakin Research, Inc.
ASDG , Inc.
Associated Computer Services
Boston Computer Society
Brookfield Communications, Inc.
Brown-Wagh Publishing
Byte by Byte Corporation
Central Coast Software
Commodore Amiga User International
Commodore Magazine
Comp-U-Save
Computer System Associates
Creative Computers
Crystal Innovations
DesignLab
Digital Creations
Digital Dynamics
Discovery Software International
Dr. T's Music Software, Inc.
Electronic Arts
Finally Software, Inc.
Firebird Licensees, Inc.
Fuller Computing Systems
Gold Disk, Inc.
Hash Enterprises
Hugh's Software Ranch
Hypertek/Silicon Springs
Impulse, Inc.
Infinity Software, Inc.
Info Magazine
Inner Connection, Inc.
InnoVision Technology
Interactive Softworks
Jumpdisk
Lattice, Inc.
Magnetic Media
Magnetic Music
Manx Software
Meridian Software, Inc.
Microillusions
MicroMagic, Inc.
Micron Technology, Inc.
Microsearch, Inc.
Microsmiths, Inc.
Mimetics Corporation
Mindware International
Mission Graphic Support
New Horizons
New Wave Software
NewTek
Oxxi, Inc.
PAR Software, Inc.
Prolific, Inc.
R & DL Productions
ReadySoft Inc.
R.G.B. Video Creations
Sedona Software
Soft Logik Corporation
Software Terminal
Software Visions, Inc.
Sound Quest, Inc.
Southern Technologies
Spencer Organization, Inc.
Spirit Technology Corp.
SunRize Industries
Supra Corporation
Syndesis
The Other Guys
TopDown Development, Inc.
Very Vivid, Ltd.
WordPerfect Corporation

We hope that You will Join AmiEXPO in Chicago, July 22-24 at The Hyatt Regency for three days of AMIGA Exhibitions, Seminars, and Keynotes!

NOW PRE-REGISTER BY PHONE

Call 800-32-AMIGA

☐ **Yes, I want to come to AmiEXPO - Midwest**
One day - $15 ____
Two days - $20 ____
Three days - $25 ____
___ **Friday** ___ **Saturday** ___ **Sunday**

Make Check of Money Order Payable to:
AmiEXPO
211 East 43rd Street, Suite 301
New York, NY 10017

NAME ____________
COMPANY ____________
ADDRESS ____________
____
CITY ____________ STATE ______ ZIP ______

For ___ MasterCard or ___ VISA Payment
Expiration Date ____________
Account Number ____________
Name as it appears on card: ____________
Signature ____________

The AMICUS & Fred Fish Public Domain Software Library

This software is collected from user groups and electronic bulletin boards around the nation. Each Amicus disk is nearly full, and is fully accessible from the Workbench. If source code is provided for any program, then the executable version is also present. This means that you don't need the C compiler to run these programs. An exception is granted for those programs only of use to people who own a C compiler.

The Fred Fish disk are collected by Mr. Fred Fish, a good and active friend of the Amiga.

Note: Each description line below may include something like 'S-O-E-D', which stands for 'source, object file, executable and documentation'. Any combination of these letters indicates what forms of the program are present. Basic programs are presented entirely in source code format.

AMICUS Disk 1

ABasic programs: Graphics

3DSolids	3d solids modeling prog. w/sample data files
Blocks	draws blocks
Cubes	draws cubes
Durer	draws pictures in the style of Durer
FScape	draws fractal landscapes
Hidden	3D drawing program, w/ hidden line removal
JPad	simple paint program
Optical	draw several optical illusions
PaintBox	simple paint program
Shuttle	draws the Shuttle in 3d wireframe
SpaceArt	graphics demo
Speaker	speech utility
Sphere	draws spheres
Spiral	draws color spirals
ThreeDee	3d function plots
Topography	artificial topography
Wheels	draws circle graphics
Xenos	draws fractal planet landscapes

ABasic programs: Tools

AddressBook	simple database program for addresses
CardFile	simple card file database program
Demo	multiwindow demo
KeyCodes	shows keycodes for a key you press
Menu	run many ABasic programs from a menu
MoreColors	way to get more colors on the screen at once, using aliasing
shapes	simple color shape designer SpeakIt speech and narrator demo

ABasic programs: Games

BrickOut	classic computer brick wall game
Othello	also known as 'go'
Saucer	simple shoot-em-up game
Spelling	simple talking spelling game
ToyBox	selectable graphics demo

ABasic programs: Sounds

Entertainer	plays that tune
HAL9000	pretends it's a real computer
Police	simple police siren sound
SugarPlum	plays "The Dance of the Sugarplum Fairies"

C programs:

ATerm	simple terminal program, S-E
cc	aid to compiling with Lattice C
decvnt	opposite of CONVERT for cross developers
Dotty	source code to the 'dotty window demo
echox	unix-style filename expansion, partial S,O-D
fasterfp	explains use of fast-floating point math
FixDate	fixes future dates on all files on a disk, S-E
freedraw	simple Workbench drawing prog.,S-E
GfxMem	graphic memory usage indicator, S-E
Grep	searches for a given string in a file with
docs.	ham shows off the hold-and-modify method of color generation
IBM2Amiga	fast parallel cable transfers between an IBM and an Amiga
Mandel	Mandelbrot set program, S-E
moire	patterned graphic demo, S-E
objfix	makes Lattice C object file symbols visible to Wack, S-E
quick	quick sort strings routine
raw	example sample window I/O
setlace	turns on interlace mode, S-E
sparks	qix-type graphic demo, S-E

Other executable programs:

SpeechToy	speech demonstration
WhichFont	displays all available fonts

Texts:

68020	describes 68020 speedup board from CSA
Aliases	explains uses of the ASSIGN command
Bugs	known bug list in Lattice C 3.02
CLICard	reference card for AmigaDOS CLI
CLICommands	guide to using the CLI
Commands	shorter guide to AmigaDOS CLI commands
EdCommands	guide to the ED editor
Filenames	AmigaDOS filename wildcard conventions
HalfBright	explains rare graphics chips that can do more colors
ModemPins	description of the serial port pinout
RAMdisks	tips on setting up your RAM: disk
ROMWack	tips on using ROMWack
Sounds	explanation of Instrument demo sound file format
Speed	refutation of Amiga's CPU and custom chip speed
WackCmds	tips on using Wack

AMICUS Disk 2

C programs:

alib	AmigaDOS object library manager , S-E
ar	text file archive program, S-E
fixobj	auto-chops executable files
shell	simple CLI shell, S-E
sq, usq	file compression programs, S-E
YachtC	a familiar game, S-E
Make	a simple 'make' programming utility, S-E
Emacs	an early version of the Amiga text editor, S-E-D

Assembler programs:

bsearch.asm	binary search code
qsort.asm	Unix compatible qsort() function, source and C test program
setjmp.asm	setjmp() code for Lattice 3.02
SVprintf	Unix system V compatible printf()
trees.o	Unix compatible tree() function, O-D

(This disk formerly had IFF specification files and examples. Since this spec is constantly updated, the IFF spec files have been moved to their own disk in the AMICUS collection.)

John Draper Amiga Tutorials:

Animate	describes animation algorithms
Gadgets	tutorial on gadgets
Menus	learn about Intuition menus

AMICUS Disk 3

C programs:

Xref	a C cross-reference gen., S-E
8bitcolor	extra-half-bright chip gfx demo, S-E
Chop	truncate (chop) files down to size, S-E
Cleanup	removes strange characters from text files
CR2LF	converts carriage returns to line feeds in Amiga files, S-E
Error	adds compile errors to a C file, S
Hello	window ex. from the RKM, S
Kermit	generic Kermit implementation, flakey, no terminal mode, S-E
Scales	sound demo plays scales, S-E
SkewB	Rubik cube demo in hi-res colors, S-E

AmigaBasicProgs(dir)

Automata	cellular automata simulation
CrazyEights	card game
Graph	function graphing programs
WitchingHour	a game

ABasiC programs:

Casino	games of poker, blackjack, dice, and craps
Gomoku	also known as 'othello'
Sabotage	sort of an adventure game

Executable programs:

Disassem	a 68000 disassembler, E-D
DpSlide	shows a given set of IFF pictures, E-D
Arrange	a text formatting program, E-D

Assembler programs:

Argoterm	terminal program with speech and Xmodem, S-E

AMICUS Disk 4 Files from the original Amiga Technical BBS

Note that some of these files are old, and refer to older versions of the operating system. These files came from the Sun system that served as Amiga technical support HQ for most of 1985. These files do not carry a warranty, and are for educational purposes only. Of course, that's not to say they don't work.

Complete and nearly up-to-date C source to 'image.ed', an early version of the Icon Editor. This is a little flaky, but compiles and runs.

An Intuition demo, in full C source, including files: demomenu.c, demomenu2.c, demoreq.c, getascii.c, idemo.c, idemo.guide, idemo.make, idemoall.h, nodos.c, and towrite.c

addmem.c	add external memory to the system
bobtest.c	example of BOB use
consoleIO.c	console IO example
creaport.c	create and delete ports
creastdio.c	create standard I/O requests
createtask.c	creating task examples
diskio.c	example of track read and write
dotty.c	source to the 'dotty window' demo
dualplay.c	dual playfield example
flood.c	flood fill example
freemap.c	old version of 'freemap'
geltools.c	tools for VSprites and BOBs
gfxmem.c	graphic memory usage indicator
hello.c	window example from RKM
inputdev.c	adding an input handler to the input stream
joystik.c	reading the joystick
keybd.c	direct keyboard reading
layertest.c	layers examples
mousport.c	test mouse port
ownlib.c,	
ownlib.asm	example of making your own library with Lattice
paratest.c	tests parallel port commands
seritest.c	tests serial port commands
serisamp.c	example of serial port use
prinintr.c	sample printer interface code
prtbase.h	printer device definitions
regiontest.c	region test program
setlace.c	source to interlace on/off program
setparallel.c	set the attributes of the parallel port
SetSerial.c	set the attributes (parity, databits)of the
singplay.c	single playfield example
speechtoy.c	source to narrator and phonetics demo
timedely.c	simple timer demo
timer.c	exec support timer functions
timrstuf.c	more exec support timer functions
WhichFont.c	loads and displays all available system fonts

process.i and prtbase.i assmebler include files:

autorqstr.txt	warnings of deadlocks with autorequesters
consoleIO.txt	copy of the RKM console I/O chapter
diskfont.txt	warning of disk font loading bug
fullfunc.txt	list of #defines, macros, functions
inputdev.txt	preliminary copy of the input device chapter

License information on Workbench distribution license

printer pre-release copy of the chapter on printer drivers, from RKM 1.1 v11fd.txt 'diff' of .fd file changes from version 1.0 to 1.1

v28v1.diff 'diff' of include file changes from version 28 to 1.0

AMICUS Disk 5 Files from the Amiga Link / Amiga Information Network

Note that some of these files are old, and refer to older versions of the operating system. These files are from AmigaLink. For a time, Commodore supported Amiga Link, aka AIN, for online developer technical support. It was only up and running for several weeks. These files do not carry a warranty, and are for educational purposes only. Of course, that's not to say they don't work.

A demo of Intuition menus called 'menudemo', in C source

whereis.c	find a file searching all subdirectories
bobtest.c	BOB programming example
sweep.c	sound synthesis example

Assembler files:

mydev.asm	sample device driver
mylib.asm	sample library example
mylib.i	
mydev.i	
asmsupp.i	
macros.i	assembler include files

Texts:

amigatricks	tips on CLI commands
extdisk	external disk specification
gameport	game port spec
parallel	parallel port spec
serial	serial port spec
v1.1update	list of new features in version 1.1
v1.1h.txt	'diff' of include file changes from version

Files for building your own printer drivers, including dospecial.c, epsondata.c, init.asm, printer.c, printer.link, printertag.asm, render.c, and wait.asm. This disk does contain a number of files describing the IFF specification. These are not the latest and greatest files, but remain here for historical purposes. They include text files and C source examples. The latest IFF spec is elsewhere in this library.

AMICUS Disk 6 IFF Pictures

This disk includes the DPSlide program, which can view a given series of IFF pictures, and the 'showpic' program, which can view each file at the click of an icon. The pictures include a screen from ArticFox, a Degas dancer, the guys at Electronic Arts, a gorilla, horses, King Tut, a lighthouse, a screen from Marble Madness, the Bugs Bunny Martian, a still from an old movie, the Dire Straits moving company, a screen from Pinball Contruction Set, a TV newcaster, the PaintCan, a world map, a Porsche, a shuttle mission patch, a tyrannosaurus rex, a planet view, a VISA card, and a ten-speed.

AMICUS Disk 7 DigiView HAM demo picture disk

This disk has pictures from the DigiView hold-and-modify video digitizer. It includes the ladies with pencils and lollypops, the young girl, the bulldozer, the horse and buggy, the Byte cover, the dictionary page, the robot and Robert. This includes a program to view each picture separately, and all together as separate, slidable screens. The 'seeilbm' program, to turn any screen into an IFF picture.

AMICUS Disk 8

C programs:

Browse	view text files on a disk, using menus S-E-D
Crunch	removes comments and white space from C files, S-E
IconExec	EXECUTE a series of commands from Workbench S-E
PDScreen Dump	dumps Rastport of highest screen to printer
SetAlternate	sets a second image for an icon, when clicked once S-E
SetWindow	makes windows for a CLI program to run under Workbench S-E
SmallClock	a small digital clockin a window menu bar
Scrimper	the screen printer in the fourth AC S-E

Amiga Basic Programs:

(Note: Many of these programs are present on AMICUS Disk 1. Several of these were converted to Amiga Basic, and are included here.)

AddressBook	a simple address book database
Ball	draws a ball
Cload	program to convert Compuserve hex files to binary, S-D
Clue	the game, Intuition driven
ColorArt	art drawing program
DeluxeDraw	the drawing program in the 3rd AC, S-D
Eliza	conversational computer psychologist
Othello	the game, as known as 'go'
RatMaze	3D ratmaze game
ROR	boggling graphics demo
Shuttle	draws 3D pictures of the space shuttle
Spelling	simple spelling program
YoYo	wierd zero-gravity yo-yo demo, tracks yo-yo to the mouse

Executable programs:

3Dcube	Modula-2 demo of a rotating cube
AltIcon	sets a second icon image, displayed when the icon is clicked
AmigaSpell	a slow but simple spell checker, E-D
arc	the ARC file compression progra must-have for telecom, E-D
Bertrand	graphics demo
disksalvage	prog. to rescue trashed disks, E-D
KwikCopy	a quick but nasty disk copy program: ignores errors, E-D
LibDir	lists hunks in an object file E-D
SaveILBM	saves any screen as IFF pic.E-D ??
ScreenDump	shareware screen dump prog, E only
StarTerm	version 2.0, term program, XmodemE-D

Texts:

LatticeMain	tips on fixing _main.c in Lattice
GDiskDrive	make your own 5 1/4 drive
GuruMed	explains the Guru numbers
Lat3.03bugs	bug list of Lattice C version 3.03
MForgeRev	user's view of the MicroForge HD
PrintSpooler	EXECUTE-based print spool prog.

.BMAP files:

These are the necessary links between Amiga Basic and the system libraries. To take advantage of the Amiga's capabilities in Basic, you need these files. BMAPs are included for 'dlist', 'console', 'diskfont', 'exec', 'icon', 'intuition', 'layers', 'mathffp', mathieeedoubas', 'mathieeesingbas', 'mathtrans', 'potgo', 'timer' and 'translator'.

AMICUS Disk 9

Amiga Basic Programs:

FlightSim	simple flight simulator program
HuePalette	explains Hue, Saturation, & Intensity
Requester	ex. of requesters from Amiga Basic
ScrollDemo	demonstrates scrolling capabilities
Synthesizer	sound program
WorldMap	draws a map of the world

Executable programs:

Boing!	latest Boing! demo,with selectable speed,E
Brush2C	converts an IFF brush to C data instructions, initialization code, E
Brush2Icon	converts IFF brush to an icon, E
Dazzle	graphics demo, tracks to mouse, E
DeciGEL	assembler program for stopping 68010 errors, S-E-D
Klock	menu-bar clock and date display, E
life	the game of life, E
TimeSet	Intuition-based way to set the time & date
EMEmacs	another Emacs, more oriented to word processing, S-E-D
MyCLI	a CLI shell, works without the Workbench, S-E-D

Texts:

FnctnKeys	read function keys from Amiga Basic
HackerSin	explains how to win the game 'hacker'
Ist68010	guide to installing a 68010 in your Amiga
Boing!	latest Boing! demo,with selectablespeed, E
Brush2C	converts an IFF brush to C data instructions, initialization code, E
Brush2Icon	converts IFF brush to an icon, E
Dazzle	graphics demo, tracks to mouse, E
DeciGEL	assembler program for stopping 68010 errors, S-E-D
Klock	menu-bar clock and date display, E
life	the game of life, E
TimeSet	Intuition-based way to set the time date,
EMEmacs	another Emacs, more oriented to word processing, S-E-D
MyCLI	a CLI shell, works without the Workbench, S-E-D

For PDS orders, please use form on page 127

Texts:

FnctnKeys	explains how to read function keys from Amiga Basic
HackerSin	explains how to win the game 'hacker'
Ist68010	guide to installing a 68010 in your Amiga
PrinterTip	sending escape sequences to your printer
StartupTip	tips on setting up your startup-sequence file
XfmrReview	list of Transformer programs that work

Printer Drivers:

Printer drivers for the Canon PJ-1080A, the C Itoh Prowriter, an improved Epson driver that eliminates streaking, the Epson LQ-800, the Gemini Star-10, the NEC 8025A, the Okidata ML-92, the Panasonic KX-P10xx family, and the Smith-Corona D300, with a document describing the installation process.

AMICUS Disk 10 Instrument sound demos

This is an icon-driven demo, circulated to many dealers. It includes the sounds of an acoustic guitar, an alarm, a banjo, a bass guitar, a boink, a calliope, a car horn, claves, water drip, electric guitar, a flute, a harp arpegio, a kickdrum, a marimba, a organ minor chord, people talking, pigs, a pipe organ, a Rhodes piano, a saxophone, a sitar, a snare drum, a steel drum, bells, a vibrophone, a violin, a wailing guitar, a horse whinny, and a whistle.

AMICUS Disk 11

C programs

dirutil	Intuition-based, CLI replacement manager S-E
cpri	shows and adjusts priority of CLI processes, S-E
ps	shows info on CLI processes, S-E
vidtex	displays Compuserve RLE pics, S-E

AmigaBasic programs

pointered	pointer and sprite editor program
optimize	optimization ex ample from AC article
calendar	large, animated calendar, diary and date book program
amortize	loan amortizations
brushtoBOB	converts small IFF brushes to AmigaBasic BOB OBJECTS
grids	draw and play waveforms
hilbert	draws Hilbert curves
madlib	mad lib story generator
mailtalk	talking mailing list program
meadows3D	3D graphics program, from A C™ article
mousetrack	mouse tracking example in hires mode
slot	slot machine game
tictactoe	the game
switch	pachinko-like game
weird	makes strange sounds

Executable programs

cp	unix-like copy command, E
cls	screen clear, S-E
diff	unix-like stream editor uses 'diff' output to fix files
pm	chart recorder performances indicator

Assembler programs

cls	screen clear and CLI arguments example

Modula-2

trails	moving-worm graphics demo
caseconvert	converts Modula-2 keywords to uppercase
Forth	Breshehan circle algorithm example
Analyze	12 templates for the spreadsheet Analyze

There are four programs here that read Commodore 64 picture files. They can translate Koala Pad, Doodle, Print Shop and News Room graphics to IFF format. Getting the files from your C-64 to your Amiga is the hard part.

AMICUS Disk 12

Executable programs

blink	'alink' compatible linker, but faster, E-D
clean	spins the disk for disk cleaners, E-D
epsonset	sends Epson settings to PAR from menu E-D
showbig	view hi-res pics in low-res superbitmap, E-D
speaktime	tell the time, E-D
undelete	undeletes a file, E-D
cnvapidhm	converts Apple][low, medium and high res pictures to IFF, E-D
menued	menu editor produces C code for menus, E-D
quick	quick disk-to-disk nibble copier, E-D
quickEA	copies Electronic Arts disks, removes protection, E-D
txed 1.3	demo of text editor from Microsmiths,E-D

C programs

spin3	rotating blocks graphics demo, S-E-D
popcli	start a new CLI at the press of a button, like Sidekick, S-E-D
vsprite	VSprite example code from Commodore, S-E-D
AmigaBBS	Amiga Basic bulletin board prog., S-D

Assembler programs

star10	makes star fields like Star Trek intro,S-E-D

Pictures

Mount Mandelbrot	3D view of Mandelbrot set
Star Destroyer	hi-res Star Wars starship
Robot	robot arm grabbing a cylinder

Texts

vendors	Amiga vendors, names, addresses
cardco	fixes to early Cardco memory boards
cinclude	cross-reference to C include files
mindwalker	clues to playing the game well
slideshow	make your own slideshows from the Kaleidoscope disk

AMICUS Disk 13

Amiga Basic programs

Routines from Carolyn Scheppner of CBM Tech Support, to read and display IFF pictures from Amiga Basic. With documentation. Also included is a program to do screen prints in Amiga Basic, and the newest BMAP files, with a corrected ConvertFD program. With example pictures, and the SaveILBM screen capture program.

Routines to load and play FutureSound and IFF sound files from Amiga Basic, by John Foust for Applied Visions. With documentation and C and assembler source for writing your own libraries, and interfacing C to assembler in libraries. With example sound.

Executable programs

gravity	Sci Amer Jan 86 gravitation graphic simulation, S-E-D

Texts

MIDI	make your own MIDI instrument interface, with documentation and a hi-res schematic picture.

AMICUS Disk 14

Several programs from Amazing Computing issues:

Tools

Dan Kary's	C structure index program, S-E-D

Amiga Basic programs:

BMAP Reader	by Tim Jones
IFFBrush2BOB	by Mike Swinger
AutoRequester	example
DOSHelper	Windowed help system for CLI commands, S-E-D
PETrans	translates PET ASCII files to ASCII files, S-E-D
C Squared	Graphics program from Scientific American, Sept 86, S-E-D
crlf	adds or removes carriage returns from files, S-E-D
dpdecode	decrypts Deluxe Paint, remo
ves copy	protection, E-D
queryWB	asks Yes or No from the user returns exit code, S-E
vc	VisiCalc type spreadsheet, no mouse control, E-D
view	views text files with window and
slider	gadget, E-D

Oing, Sproing, yaBoing, Zoing are sprite-based

Boing! style	demos, S-E-D

CLIClock, sClock, wClock are window border clocks, S-E-D

Texts

An article on long-persistance phospor monitors, tips on making brushes of odd shapes in Deluxe Paint, and recommendations on icon interfaces from Commodore-Amiga.

AMICUS Disk 15

The C programs Include:

'pr'	a file printing utility, which can print files in the background, and with line numbers and control character filtering.
'fm'	displays a chart of the blocks allocated on a disk.
'Ask'	questions an 'execute' file, returns an error code to control the execution in that batch file
'Stat'	an enhanced version of AmigaDOS 'status' command.
'Dissolve'	random-dot dissolve demo displays IFF picture slowly, dot by dot, in a random fashion.
'PopCLI2'	invoke new CLI window at the press of a key.

The executable programs Include:

'Form'	file formatting program through the printer driver to select print styles
'DiskCat'	catalogs disks, maintains, sorts,merges lists of disk files
'PSound'	SunRize Industries' sampled sound editor & recorder
'iconmaker'	makes icons for most programs
'Fractals'	draws great fractal seascapes and mountain scapes.
'3D Breakout'	3D glasses, create breakout in a new dimension
'AmigaMonitor'	displays lists of open files,
memory use,	tasks, devices and ports in use.
'Cosmoroids'	version of 'asteroids' for the Amiga.
'Sizzlers'	high resolution graphics demo written in Modula 2.

Texts:

'ansi.txt'	explains escape sequences the CON: device responds to.
'FKey'	includes template for making paper to sit in the tray at the top of the Amiga keyboard.
'Spawn'	programmer's document from Commodore

Amiga, describe ways to use the Amiga's multitasking capabilities in your own programs.

AmigaBasic programs:

'Grids'	draw sound waveforms, and hear them played.
'Light'	a version of the Tron light-cycle video game.
'MigaSol'	a game of solitaire.
'Stats'	program to calculate batting averages
'Money'	"try to grab all the bags of money that you can."

AMICUS 15 also includes two beautiful IFF pictures, of the enemy walkers from the ice planet in Star Wars, and a picture of a cheetah.

AMICUS Disk 16

'juggler' demo by Eric Graham. a robot juggler bouncing three mirrored balls, with sound effects. Twenty-four frames of HAM animation are flipped quickly to produce this image. You control the speed of the juggling. The author's documentation hints that this program might someday be available as a product.

IFF pictures

parodies of the covers of Amiga World and Amazing Computing magazines.

C programs:

'inputhandler'	example of making an input handler.
'FileZap3'	binary file editing program
'ShowPrint'	displays IFF picture, and prints it.
'Gen'	program indexes and retrieves C structures and variables declared in the Amiga include file system.

Executable Programs:

'FixHunk2'	repairs an executable program file for expanded memory
'ms2smus'	converts Music Studio files to IFF standard 'SMUS' format. I have heard this program might have a few bugs, especially in regards to very long songs, but it works in most cases.
'Missile'	Amiga version of the 'Missile Command' video game,

This disk also contains several files of scenarios for Amiga Flight Simulator II. By putting one of these seven files on a blank disk, and inserting it in the drive after performing a special command in this game, a number of interesting locations are preset into the Flight Simulator program. For example, one scenario places your plane on Alcatraz, while another puts you in Central Park

AMICUS Disk 17

Telcommunications disk whuch contains six terminal programs.

"Comm" V1.33	term prog. with Xmodem, WXmodem,
"ATerm" V7.2	term prog. includes Super Kermit
"VT-100"V2.6	Dave Wecker's VT-100 emulator with Xmodem,Kermit, and scripting
"Amiga Kermit"	V4D(060) port of the Unix C-Kermit
"VTek" V2.3.1	Tektronix graphics terminal emulator based on the VT-100 prog. V2.3 and contains latest 'arc' file compression
"AmigaHost"	V0.9 for Compuserve. Includes RLE graphics abilities & CIS-B file transfer protocol.
"FixHunk"	expansion memory necessity
"FixObj"	removes garbage characters from modem received files
"Txt"	filters text files from other systems to be read by the Amiga E.C.
"addmem"	executeable version for use with mem expansion article in AC v2..1
'arc'	file documentation and a basic tutorial on un 'arcing files
"arcre"	for makeing "arc" files E.C.

AMICUS Disk 18

Logo	Amiga version of the populacomputer language, with example programs, E-D
Tv*Text	Demo version of the TV*Text character generator
PageSetter	Freely distributable versions of the updated PagePrint and PageIFF programs for the PageSetter desktop publishing package.
FullWindow	Resizes any CLI window using only CLI commands, E-D
Life3d	3-D version of Conway's LIFE program, E-D
Defdisk	CLI utility to re-assign a new Workbench disk, S-E-D
Calendar.WKS	Lotus-compatible worksheet that makes calendars
SetKey	Demo of keyboard key re-programmer, with IFF picture to make function key labels, E-D
VPG	Video pattern generator for aligning monitors, E-D
HP-10C	Hewlett-Packard-like calculator, E-D
SetPrefs	Change the Preferences settings on the fly, in C, S-E-D
StarProbe	Program studies stellar evolution. C source included for Amiga and MS-DOS, S-E-D
ROT	C version of Colin French's AmigaBasic ROT program from Amazing Computing. ROT edits and displays polygons to create three dimensional objects. Up to 24 frames of animation can be created and displayed. E-D
Scat	Like Ing, windows on screen run away from the mouse, E-D
DK	Decays" the CLI window into dust, in Modula 2, S-E-D
DropShadow2	Adds layered shadows to Workbench windows, E-D

AMICUS Disk 19

This disk carries several programs from Amazing Computing. The IFF pictures on this disk include the Amiga Wake part T-shirt logo, a sixteen-color hi-res image of Andy Griffith, and five Amiga Live! pictures from the Amazing Stories episode that featured the Amiga.

Solve	Linear equation solver in assembly language, S-E-D
Gadgets	Bryan Catley's AmigaBasic tutorial,
Household	Bryan Catley's AmigaBasic household inventory program, S-D
Waveform	Jim Shields' Waveform WoAmigaBasic, S-D
DiskLib	John Kennan's AmigaBasic disk librarian program, S-D
Subscripts	Ivan Smith's AmigaBasic subscript example, S-D
String, Boolean	C programs and executables for Harriet Maybeck Tolly's Intuition tutorials, S-E-D
Skinny C	Bob Riemersma's example for making small C programs, S-E-D
COMAL.h	Make C look like COMAL Header file,
EmacsKey	Makes Emacs function key definitons by Greg Douglas, S-D
AMon 1.1	Snoop on system resource use, E-D
BTE	Bard's Tale character editor, E-D
Size	CLI program shows the size of a given set of files, E-D
WinSize	CLI window utility resizes current window, S-E-D

AMICUS Disk 20

Compactor, Decoder Steve Michel AmigaBasic tools, S-D

BobEd	BOB and sprite editor written in C,S-E-D
SpriteMasterII	Sprite editor and animator by Brad Kiefer, E-D
BlitLab	Blitter chip exploration C program by Tomas Rokicki, S-E-D
FPic	Image processing program by Bob Bush loads and saves IFF images, changes them with several techniques, E-D
Bankn	Complete home banking program, balance your checkbook! E-D

AMICUS Disk 21

Target	Makes each mouse click sound like a gunshot, S-E-D
Sand	Simple game of sand that follows the mouse pointer, E-D
PropGadget	Harriet Maybeck Tolly's proportional gadget example, S-E
EHB	Checks to see if you have extra-half-bright graphics, S-E-D
Piano	Simple piano sound program
CelScripts	Makes cel animation scripts for Aegis Animator, in AmigaBasic

This disk has electronic catalogs for AMICUS disks 1 to 20 and Fish disks 1 to 80. They are viewed with the DiskCat program, included here.

AMICUS Disk 22

Cycles	Light cycle game, E-D
Show_PrintII	Views and prints IFF pictures, including larger than screen
PrtDrvGen2.3	Latest version of a printer driver generator
Animations	VideoScape animations of planes and boing ball
Garden	Makes fractal gardenscapes
BasicSorts	Examples of binary search and insertion sort in AmigaBasic

AMICUS Disk 23

An AMICUS disk	completely dedicated to music on the Amiga. This disk contains two music players, songs, instruments, and players to bring the thrill of playing "Big Sound" on your Amiga
Instruments	a collection of 25 instruments for playing and creating music. The collection ranges from Cannon to Marimba
List INSTR	program to list the instruments DMCS will not load as well as list the origins for any instrument.
Music	a collection of 14 Classical pieces
1812Overture	The 16 minute classical feature complete with Cannon!

Three Amiga Music Players:

SMUSPlay
MusicCraft2SMUS
MusicStudio2SMUS

AMICUS Disk 24

Sectorama	A disk sector editor for any AmigaDOS file-structured device, recover files from a trashed hard disk. By David Joiner of MicroIllusions
Iconize	Reduces the size of IFF images, companion program, Recolor, remaps the palette colors of one picture to use the palette colors of another. Using these programs and a tool to convert IFF brushes to Workbench icons, make icons look like miniatures of the pictures.
CodeDemo	Modula-2 program converts assembler object files to inline CODE statements. Comes with a screen scrolling example
AmiBug	Workbench hack makes the same fly walk across the screen at random intervals. Otherwise, completely harmless.
BNTools	Three examples of assembly language code from Bryce Nesbitt: 1. SetLace,prog to switch interlace on&off. 2. Why, replace AmigaDOS CLI Why 3. LoadIt, prog to load a file into memory until a reboot. (Only the most esoteric hackers will find LoadIt useful.)
Monolace	CLI program resets Preferences to several colors of monochrome & interlace screens. C source is included, works with DisplayPref, a CLI program which displays the current Preferences settings.
BoingMachine	A ray-traced animation of a perpetual motion Boing-making machine, includes the latest version of the Movie program, which has the ability to play sounds along with the animation. By Ken Offer
Daisy	Example of using the translator and narrator devices to make the Amiga talk. It is written in C.
QuickFlix	Script-driven animation and slideshow program flips through IFF images.
BMon	System monitor AmigaBasic program ; perform simple manipulations of memory.
Moose	Random background program, a small window opens with a moose resembleing Bullwinklesaying witty phrases user definable.
DGCS	Deluxe Grocery Construction Set, simple Intuition-based prog for assembling and printing a grocery list.
The Virus Check	directory holds several programs relating to the software virus that came to the US from pirates in Europe as detailed in Amazing Computing V2.12. Bill Koester's full explanation of the virus code is included. One program checks for the software virus on a Workbench disk; the second program checks for the virus in memory, which could infect other disks.

AMICUS Disk 25

Nemesis	Graphics demo pans through space towards the mythical dark twin of the sun with wonderful music and space graphics.
The KickPlay	directory holds text that describes several patches to the Kickstart disk. For Amiga 1000 hackers who feel comfortable patching a disk in hexadecimal, KickPlay offers the chance to automatically do an ADDMEM for old expansion memory, as well as the ability to change the picture of the "Insert Workbench" hand. A program is also included for restoring the correct checksum of the Kickstart disk.
KeyBird	BASIC prog edits keymaps, adjust the Workbench keymaps or create your own.

8ColorWB	Modifies the Workbench so three bitplanes are used, icons can have eight colors, instead of four, eight-color icons are included. Public domain program 'zapicon' or 'brush2icon' converts eight-color IFF brushes to icons, to use Deluxe Paint to make icons for this new Workbench.
BrushIcon	Converts brushes to icons (bizarr docs).
Egraph	Graphing prog reads [x,y] values from a file and displays them on the screen, similar to the same-named Unix program.
Keep 1.1	Message-managing program for telecommunications, lets you save messages from an online transcript to another file, understands the message format of the national networks and several types of bulletin board software. Moves through the transcript and save messages
Kill.fastdir	Speed up directory access, it creates a small file in each directory on a disk which contains the information about the files, will also remove all the 'fastdir' files from each directory. by CLimate's authors

The LaceWB program changes between interlace and non-interlace Workbench. Previously, you were forced to reboot after changing Preferences to an interlaced screen. This program flips between the normal and extended screen heights.

PW_Utility	A shareware utility for ProWrite users, changes margin settings and font types.
Guru	A CLI program, prints out probable causes for Guru mediations; C source included.
DiskWipe	Latest from Software Distillery, removes files from directories or disk drives, much faster than 'delete.'
Snow	AmigaBasic makes snowflake designs.
Mlist	Mailing list database.
Softballstats	Maintain softball statistics/ team records.
Dodge	Short Modula-2 program moves the Workbench screen around after a period of time, prevents monitor burn-in.

AMICUS Disk 26

Todor Fay's SoundScape module code from his Amazing Computing articles. The source to Echo, Chord, TX, and VU is included. The Lattice and Manx C source code is here, along with the executable modules.

ImageMaker	Interesting tool edits Image structures for C, loads & saves C code directly.
Claz2	Update of prog to convert IFF images to PostScript files for printing on laser printers
SDBackup	Hard disk backup prog with Lempel-Ziv compression to reduce the necessary number of disks.
TCB	Prints information about tasks and processes in the system; assembler source is included.
FunBut	Lets a function key act like a rapid series of left mouse button events.
DC	A handy program for people who use an Amiga 1020 5 1/4 inch drive as an AmigaDOS floppy. A Workbench program that sends a DiskChange signal to the operating system: instead of typing 'diskchange df2:' over and over again, just click on the icon. C source included.
System config	File makes screen 80 columns wide of text in the Scribble! word processor.
Dick2Ram	2 programs to move the Scribble! spelling dictionary to and from the RAM disk.
Lexical	Analyzes a text file and gives the Gunning-Fog, Flesch, and Kincaid indices which measure readability.
HexDump	Modula-2 program to display memory locations in hexadecimal.
Tartan	AmigaBasic; design Tartan plaids.
DirMaster	Disk catalog program.
BMP	plays 8SVX sampled sounds in the background while something else is happening in the Amiga, as your Amiga is booting, for example.
ShowPt	CLI program changes your pointer to a given pointer.

AMICUS 26 also has a collection of mouse pointers, & Workbench program to display them

Fred Fish Public Domain Software

Fred Fish Disk 1:

amigademo	Graphical benchmark for comparing amigas.
amigaterm	simple communications program with Xmodem
balls	simulation of the 'kinetic thingy' with balls on strings
colorful	Shows off use of hold-and-modify mode.
dhrystone	Dhrystone benchmark program.
dotty	Source to the 'dotty window' demo on the Workbench disk.
freedraw	A small 'paint' type program with lines, boxes, etc.
gad	John Draper's Gadget tutorial program
gfxmem	Graphical memory usage display prog.
halfbrite	demonstrates 'Extra-Half-Brite' mode, if you have it
hello	simple window demo
lattfp	accessing the Motorola Fast Floating Point library from C
palette	Sample prog. to design color palettes.
trackdisk	Demonstrates use of the trackdisk driver.
requesters	John Draper's requester tutorial and example program.
speech	Sample speech demo program. Stripped down 'speechtoy'.
speechtoy	Another speech demo program.

Fred Fish Disk 2:

alib	Object module librarian.
cc	Unix-like frontend for Lattice C compiler.
dbug	Macro based C debugging package. Machine independent.
make	Subset of Unix make command.
make2	Another make subset command.
microemacs	Small version of emacs editor, with macros, no extensions
portar	Portable file archiver.
xrf	DECUS C cross reference utility.

Fred Fish Disk 3:

gothic	Gothic font banner printer.
roff	A 'roff' type text formatter.
ff	A very fast text formatter
cforth	A highly portable forth implementation. Lots of goodies.
xlisp	Xlisp 1.4, not working correctly.

Fred Fish Disk 4;

banner	Prints horizontal banner
bgrep	A Boyer-Moore grep-like utility
bison	GNU Unix replacement 'yacc', not working.
bm	Another Boyer-Moore grep-like utility
grep	DECUS grep
kermit	simple portable Kermit with no connect mode.
MyCLI	Replacement CLI for the Amiga. V. 1.0
mandel	A Mandelbrot set program, by Robert French and RJ Mical

Fred Fish Disk 5

cons	Console device demo program with supporting macro routines.
freemap	Creates a visual diagram of free memory
input.dev	sample input handler, traps key or mouse events
joystick	Shows how to set up the gameport device as a joystick.
keyboard	demonstrates direct communications with the keyboard.
layers	Shows use of the layers library
mandelbrot	IFF Mandelbrot program
mouse	hooks up mouse to right joystick port
one.window	console window demo
parallel	Demonstrates access to the parallel port.
printer	opening and using the printer, does a screen dump, not working
print.support	Printer support routines, not working.
proctest	sample process creation code, not working
region	demos split drawing regions
samplefont	sample font with info on creating your own
serial	Demos the serial port
singlePlayfield	Creates 320 x 200 playfield
speechtoy	latest version of cute speech demo
speech.demo	simplified version of speechtoy, with IO requests
text.demo	displays available fonts
timer	demos timer.device use
trackdisk	demos trakcdisk driver

Fred Fish Disk 6:

compress	like Unix compress, a file squeezer
dadc	analog clock impersonator
microemacs	upgraded version of microemacs from disk 2
mult	removes multiple occuring lines in files
scales	demos using sound and audio functions
setparallel	Allows changing parallel port parameters
setserial	Allows changing serial port parameters.
sortc	quicksort based sort program, in C
stripc	Strips comments and extra whitespace from C source

Fred Fish Disk 7:

This disk contains the executables of the game Hack V 1.0.1.

Fred Fish Disk 8:

This disk contains the C source to Hack on disk 7.

Fred Fish Disk 9:

moire	Draws moire patterns in black and white
MVP-FORTH	Mountain View Press Forth, version 1.00.03A. A shareware version of FORTH from Fantasia Systems.
proff	a more powerful text formatting program
setlace	Prog to toggle interlace mode on and off
skewb	a rubic's cube type demo
sparks	moving snake Graphics demo

Fred Fish Disk 10:

conquest	An interstellar adventure simulation game
dehex	convert a hex file to binary
filezap	Patch program for any type of file.
fixobj	Strip garbage off Xmodem transferred files.
iff	Routines to read and write iff format files.
ld	simple directory program
ls	Minimal UNIX ls, with Unix-style wildcarding, in C
sq,usq	file squeeze and unsqueeze
trek73	Star Trek game
yachtc	Dice game.

Fred Fish Disk 11:

dpslide	slide show program for displaying IFF images with miscellaneous pictures

Fred Fish Disk 12:

amiga3d	Shows a rotating 3 dimensional solid 'Amiga sign'.
ArgoTerm	a terminal emulator program, written in assembler
arrow3d	Shows a rotating 3 dimensional wire frame arrow.
ld4	directory listing program
IconExec	
SetWindow	two progs for launching progs from Workbench, presently only works under CLI.
SetAlternate	Makes an icon show a second image when clicked once
StarTerm	terminal emulator, with ASCII Xmodem, dialer, more.

Fred Fish Disk 13:

A Bundle of Basic programs, including:

Jpad
xmodem
ror
bounce
card1
cubes1
dragon
Eliza
fscape
hal9000
join
minipaint
pena
Readme
sabotage
shuttle
sketchpad
speecheasy
spiral
talk
termtest
wheels
toybox
3dsolids
amgseq1
box
circle
cutpaste
draw
ezterm
gomoku
halley
loz
mouse
pinwheel
rgb
salestalk
spaceart
spell
striper
terminal
tom
xenos
ezspeak
addbook
amiga-copy
brickout
colorcircles
datedogstar
dynamictriangle
fillibuster
dart
hauntedM
mandel
Orthello
gboxrandom-circles
rgbtest
shades
speakspeach
sphere
superpad
topography
xmostriper
mandlebrot
algebra
band
canvas
Copy
fractal
haiku
hidden
menu
patch
Rord
shapes
suprshr
triangle

(note: some programs are Abasic, most are Amigabasic, and some programs are presented in both languages)

Fred Fish Disk 14;

amiga3d	update of #12, includes C source to a full hidden surface removal and 3D graphics
beep	Source for a function that generates a beep sound
dex	extracts text from within C source files
dimensions	demonstrates N dimensional graphics
filezap	update of disk 10, a file patch utility
gfxmem	update of disk 1, graphic memory usage indicator
gi	converts IFF brush files to Image struct, in C text.
pdterm	simple ANSI VT100 terminal emulator, in 80 x 25 screen
shell	simple Unix 'csh' style shell
termcap	mostly Unix compatible 'termcap' implementation.

Fred Fish Disk 15:

Blobs	graphics demo, like Unix 'worms'
Clock	simple digital clock program for the title bar
Dazzle	An eight-fold symmetry dazzler program. Really pretty!
Fish	double buffered sequence cycle animation of a fish
Monopoly	A really nice monopoly game written in AbasiC.
OkidataDump	Okidata ML92 driver and WorkBench screen dump program.
Polydraw	A drawing program written in AbasiC.
Polyfractals	A fractal program written in AbasiC.

Fred Fish Disk 16:

A complete copy of the latest developer IFF disk

Fred Fish Disk 17:

The NewTek Digi-View video digitizer HAM demo disk

Fred Fish Disk 18:

AmigaDisplay	dumb terminal program with bell, selectable fonts
Ash	Prerelease C Shell-like shell program, history, loops, etc.
Browser	wanders a file tree, displays files, all with the mouse
MC68010	docs on upgrading your Amiga to use a MC68010
Multidim	rotate an N dimensional cube with a joystick
PigLatin	SAY command that talks in Pig Latin
Scrimper	Screen image printer
Xlisp1.6	source, docs, and execut for a Lisp interpret.

Fred Fish Disk 19:

BlackJack	text-oriented blackjack game
JayMinerSlides	Slides by Jay Miner, Amiga graphics chip designer, showing flowchart of the Amiga internals, in 640 x 400.
Keymap_Test	test program to test the key mapping routines
LockMon	Find unclosed file locks, for programs that don't clean up.

Fred Fish Disk 20:

AmigaToAtari	converts Amiga object code to Atari form
DiskSalv	program to recover files from a trashed AmigaDOS disk.
Hash	example of the AmigaDOS disk hashing function
Hd	Hex dump utility ala Computer Language magazine, April 86
MandelBrots	Mandelbrot contest winners
MultiTasking	Tutorial and examples for Exec level multitasking
Pack	strips whitespace from C source
PortHandler	sample Port-Handler program that performs. Shows BCPL environment
Random	Random number generator in assembly, for C or assembler.
SetMouse2	sets the mouse port to right or left
SpeechTerm	terminal Emulator with speech capabilities, XModem
TxEd	Demo editor from Microsmith's Charlie Heath

Fred Fish Disk 21

This is a copy of Thomas Wilcox's Mandelbrot Set Explorer disk. Very good!

Fred Fish Disk 22

This disk contains two new 'strains' of microemacs.

Lemacs	version 3.6 by Daniel Lawrence. For Unix V7, BSD 4.2, Amiga, MS-DOS, VMS. Uses Amiga function keys, status line, execute, startup files, more.
Pemacs	By Andy Poggio. New features include <ALT> keys as Meta keys, mouse support, higher priority, backup files, word wrap, function keys.

Fred Fish Disk 23

Disk of source for MicroEmacs, several versions for most popular operating systems on micros and mainframes. For people who want to port MicroEmacs to their favorite machine.

Fred Fish Disk 24:

Conques	interstaller adventure simulation game
Csh	update to shell on Disk 14, with built in commands,named variables substitution.
Modula-2	A pre-release version of the single pass Modula-2 compiler originally developed for Macintosh at ETHZ. This code was transmitted to the AMIGA and is executed on the AMIGA with a special loader. Binary only.

Fred Fish Disk 25

Graphic Hack	A graphic version of the game on disks 7 and 8 This is the graphics-oriented Hack game by John Toebes. Only the executable is present.

Fred Fish Disk 26

UnHunk	Processes the Amiga 'hunk' loadfiles. Collect code, data, and bss hunks together, allows individual specificatio of code, data, and bss origins, and generates binary file with format reminiscent of Unix 'a.out' format. The output file can be easily processed by a separate program to produce Motorola 'S-records' suitable for downloading to PROM programmer. By Eric Black.
C-kermit	Port of the Kermit file transfer program and server.
Ps	Display and set process priorities
Archx	Yet another program for bundling up text files and mailing or posting them as a single file unit.

Fred Fish Disk 27

ABdemos	Amiga Basic demos; Caroly Scheppner.
NewConvertFD	creates .bmaps from fd files.
BitPlanes	finds addresses of and writes to bitplanes of the screen's bitmap.
AboutBmaps	A tutorial on creation and use of bmaps.
LoadILBM	loads and displays IFF ILBM pics.
LoadACBM	loads and displays ACBM pics.
ScreenPrint	creates a demo screen and dumps it to a graphic printer.
Disassem	Simple 68000 disassembler. Reads standard Amiga object files and disassembles the code sections. Data sections are dumped in hex. The actual disassember routines are set up to be callable from a user prog so instructions in memory can be disassembled dynamically. By Bill Rogers.
DvorakKeymap	Example of a keymap structure for the Dvorak keyboard layout. Untested but included because assembly examples are few and far between. By Robert Burns
Hypocycloids	Spirograph, from Feb. 84 Byte.
LinesDemo	Example of proportional gadgets to scroll a SuperBitMap.
MemExpansion	Schematics and directions for building your own homebrew 1 Mb memory expansion, by Michael Fellinger.
SafeMalloc	Program to debug 'malloc()' calls
ScienceDemos	Convert Julian to solar and sidereal time, stellar positions and radial velocity epoch calculations and Galilean satellite plotter. By David Eagle.

Fred Fish Disk 28

ABasic games by David Addison: Backgammon , Cribbage, Milestone, and Othello

Cpp	DECUS 'cpp' C preprocessor, & a modified 'cc' that knows about the 'cpp', for Manx C.
Shar	Unix-compatible shell archiver, for packing files for travel.
SuperBitMap	Example of using a ScrollLayer, syncing SuperBitMaps for printing, and creating dummy RastPorts.

Fred Fish Disk 29

AegisDraw Demo	Demo program without save and no docs.
Animator Demo	Player for the Aegis Animator files
Cc	Unix-like front-end for Manx C
Enough	Tests for existence of system resources, files, and devices
Rubik	Animated Rubik's cube program
StringLib	
Vt100	VT-100 terminal emulator with Kermit and Xmodem protocols

Fred Fish Disk 30

Several shareware programs. The authors request a donation if you find their program useful, so they can write more software.

BBS	an Amiga Basic BBS by Ewan Grantham
FineArt	Amiga art
FontEditor	edit fonts, by Tim Robinson
MenuEditor	Create menus, save them as C source, by David Pehrson
StarTerm3.0	Very nice telecom by J. Nangano

(Fred Fish Disk#30 is free if requested when ordered with at least three other disks from the collection.)

Fred Fish Disk 31

Life	Life game, uses blitter to do 19.8 generations a second.
Mandelbrot	Version 3.0 of Robert French's program.
MxExample	Mutual exclusion gadget example.
RamSpeed	Measure relative RAM speed, chip and fast.
Set	Replacement for the Manx 'set' command for environment variables,with improvements.
Tree	Draws a recursive tree, green leafy type, not files.
TxEd	Crippled demo version of Microsmith's text editor, TxEd.
VDraw	Full-featured drawing program by Stephen Vermeulen.
Xicon	Invokes CLI scripts from icon
Ticon	Displays text files from an icon.

Fred Fish Disk 32

Address	Extended address book, AmigaBASIC
Calendar	Calendar/diary program, AmigaBASIC
DosPlus1	First volume of CLI oriented developer tools
DosPlus2	2nd volume of CLI oriented developer tools.
Executables only:	
MacView	Views MacPaint pics in Amiga low or high res, no sample pictures, by Scott Evernden.
Puzzle	Simulation of puzzle with moving squares.
ShowHAM	View HAM pictures from CLI.
Solitaire	ABasiC games of Canfield and Klondike, from David Addison.
Spin3	Graphics demo of spinning cubes, double-buffered example.
Sword	Sword of Fallen Angel text adventure game written in Amiga Basic.
Trails	Leaves a trail behind mouse, in Modula-2

Fred Fish Disk 33

3dstars	3d version of the "stars" program below.
Bigmap	Low-level graphics example scrolls bitmap with ScrollVPort.
Dbuf.gels	Double-buffered animation example for BOBs and VSprites.
DiskMapper	Displays sector allocation of floppy disks.
MemView	View memory in real time, move with joystick.
Oing	Bouncing balls demo
Sproing	Oing, with sound effects.
ScreenDump	Dumps highest screen or window to the printer.
Sdb	Simple database program from a DECUS tape.
Stars	Star field demo, like Star Trek.
TermPlus	Terminal program with capture, library, function keys, Xmodem, CIS-B protocols.
Vt100	Version 2.0 of Dave Wecker's VT-100 emulator, with scripts & function

Fred Fish Disk 34

Alint	Support files for Gimpel's 'lint' syntax checker
Blink	PD 'alink' compaible linker,faster,better.
Browser	Updated to FF 18 'browser', in Manx, with scroll bars, bug fixes.
Btree	b-tree data structure examples
Btree2	Another version of 'btree'
Calendar	Appointment calendar with alarm.
Less	File viewer, searching, position by percent, line number.
NewFonts	Set of 28 new Amiga fonts from Bill Fischer
Pr	Background print utility, style options, wildcards.
Requester	Deluxe Paint-type file requester, with sample.

Fred Fish Disk 35

ASendPacket	C example of making asynchronous I/O calls to a DOS handler, written by C-A
ConsoleWindow	C example of getting the Intuition pointer a CON: or RAW: window, for 1.2, by C-A.
DirUtil	Walk the directory tree, do CLI operations from menus
DirUtil2	Another variant of Dirutil.
FileRequester	Lattice C file requester module, with demo driver, from Charlie Heath.
MacView	Views MacPaint pictures in Amiga low or high res, with sample pictures, by Scott Evernden.
Plop	Simple IFF reader program
PopCLI	Sidekick-style program invokes a new CLI, with automatic screen blanking.
QuickCopy	Devenport disk copiers duplicate copy-protected disks.
ScrollPf	Dual playfield example, from C-A, shows 400 x 300 x 2 bit plane playfield on a 320 x 200 x 2 plane deep playfield.
SendPacket	General purpose subroutine to send AmigaDos packets.
SpriteMaker	Sprite editor, can save work as C data structure. Shareware by Ray Larson.
Tracker	Converts any disk into files, for electronic transmission. Preserves entire file structure. Shareware by Brad Wilson.
TriClops 3-D	space invasion game, formerly commercial, now public domain. From Geodesic Publications.
Tsize	Print total size of all files in subdirectories.
Unifdef	C preprocessor to remove given #ifdef'd sections of a file, leaving the rest alone. By Dave Yost
Vttest	VT-100 emulation test program. Requires a Unix system.

Fred Fish Disk 36

Acp	Unix-like 'cp' copy program
Clock	Updated version of clock on disk 15.
Csh	Manx 'csh'-like CLI, history, variables, etc.
DietAid	Diet planning aid organizes recipes, calories
Echo	Improved 'echo' command with color, cursor addressing
FixHunk	Fixs programs to let them run in external memory.
Fm	Maps the sectors a file uses on the disk.
KickBench	Docs, program to make a single disk that works like a Kickstart and Workbench.
Lex	Computes Fog, Flesch, and Kincaid readability of text files.
TunnelVision	David Addison ABasic 3D maze perspective game.
Vc	Visicalc-like spreadsheet calculator program.
Vt100	Version 2.2 of Dave Wecker's telecom program
YaBoing	Oing! style game program shows sprite collision detects

Fred Fish Disk 37

This disk is a port of Timothy Budd's Little Smalltalk system, done by Bill Kinnersley at Washington State University.

Fred Fish Disk 38

CSquared	Sep 86 Sci American, Circle Squared algorithm
FixObj	Strips garbage off Xmodem transfered object files
Handler	AmigaDOS handler (device) example from C-A
Hp-10c	Mimics a HP-10C calculator, written in Modula-2
IFFEncode	Saves the screen as an IFF file
IffDump	Dumps info about an IFF file
Jsh	BDS C-like CLI shell
NewStat	STATUS-like program, shows priority, processes
Reversi	Game of Reversi, version 6.1
UUdecode	Translate binary files to text, Unix-like programs
Vdraw	Drawing program, version 1.14
VoiceFiler	DX MIDI synthesizer voice filer program
Window	Example of creating a D OS window on a custom screen

Fred Fish Disk 39

AnsiEcho	'echo', 'touch', 'list', 'cls' written in assembler.
Display	Displays HAM images from a ray-tracing program, with example pictures.
Driver	Example device driver source, acts like RAM: disk
Xlisp	XLisp 1.7, executable only

Fred Fish Disk 40

Ahost	Terminal emulator with Xmodem, Kermit and CIS B protocols, function keys, scripts, RLE graphics and conference mode.
AmigaMonitor	Dynamically displays the machine state, such as open files, active tasks, resources, device states, interrupts, libraries, ports, etc.
Arc	Popular file compression system, the standard for transiting files
AreaCode	Program that decodes area codes into state and locality.
Blink	'alink' replacement linker, version 6.5
Cosmo	An 'asteriods' clone.
Dg210	Data General D-210 Terminal emulator
DirUtil	Windowed DOS interface program, V 1.4
DOSHelper	Windowed AmigaDOS CLI help program
PagePrint	Prints text files with headers, page breaks, line numbers
PopCLI	Starts a new CLI with a single keystroke, from any program, With a screen-saver feature. Version 2, wisource.
SpriteEd	Sprite Editor edits two sprites at a time
X-Spell	Spelling checker allows edits to files

Fred Fish Disk 41

AmigaVenture	Create your own text adventure programs in AmigaBasic.
Csh	Version 2.03 of Dillon's C sh-like shell. Executable only
Dbug	Macro based C debugging packagepdate to FF #2
DualPlayField	example from CBM, update to Intuition manual
GetFile	Heath's file requester, with source
LatXref	Cross reference of Lattice 3.10 header files
Lines	Line drawing demo program
SetFont	Changes font used in a CLI window
Vt100	Version 2.3 of the VT-100 terminal program.

Fred Fish Disk 42

This disk contains an Amiga version of MicroGNUEmacs.

Fred Fish Disk 43

BasicBoing	AmigaBasic program demos page flipping of a 3D cube
Bbm	Demo copy of B.E.S.T. Business Management System.
BbsList	A list of Amiga Bulletin Board Systems
Cc	C compiler frontends for Manx and Lattice C
Copper	A hardware copper list disassembler
InstIFF	Converts Instruments demo sounds to IFF sampled sounds
PopColours	Adjust RGB colors of any screen
SpriteClock	Simple clock is displayed on a sprite above all screens
ST Emulator	Non-serious Atari ST emulator
WBrun	Lets Workbench programs be run from the CLI
Wild	Two Unix shell style wild card matching routines

Fred Fish Disk 44

Icons	Miscellaneous icons
NewIFF	New IFF material from CBM for sampled voice and music files
RayTracePics	The famous ray-tracing pictures, from FF#39, now converted to IFF HAM format for "much" faster viewing.
ViewILBM	Displays normal and HAM ILBM files

Fred Fish Disk 45

Clue	Clue board game
Make	Another 'make', with more features
Pictures	Miscellaneous pictures
Update	Updates older disk with newer files fromanother disk
WhereIs	Searches a disk for files of given name

Fred Fish Disk 46

Asm	Shareware 68010 macro assembler, ROM Kernal Manual compatible
CheckModem	'execute' file program detects presence of modem
Egad	Gadget editor from the Programmers Network
Jive	Transforms a file from English to Jive.
My.lib	A binary only copy of Matt's alternate runtime library. Author: Matt Dillon
ProffMacros	Subset Berkeley 'ms' and 'mm' macros for 'proff'
ValSpeak	Transforms a file from English to Valley Speak.

Fred Fish Disk 47

3D-Arm	Simulation of a robotic arm, very good graphics, teaching tool, including C source.
Juggler	Eric Graham's stunning HAM animation of a robot juggler
VT-100	Version 2.4 of Dave Wecker's terminal emulator, with Xmodem and Kermit file transfer protocols

Fred Fish Disk 48

Bru	Alpha version of a hard disk file archiver
Comm	Version 1.30 of a terminal emulator with phone directories
Csh	Version 2.04 of Matt Dillion's Unix 'csh'-like CLI replacement, including Lattice & Manx C source
Diskperf	Disk benchmark program for Unix and Amiga
Du	Computes disk storage of a file or directory
MemWatch	Program to watch for programs that trash low memory. It attempts to repair the damage, and puts up a requester to inform you of the damage. From the Software Distillery.
Profiler	A realtime execution profiler for Manx C programs. Includes C source.

Fred Fish Disk 49

Cycloids	Update of electronic spirograph from disk 27
DirUtil	Enhanced version of DirUtil from disk 35
MultiDef	Scans a set of object modules and libraries searching for multiply defined symbols
MyUpdate	Disk update utility with options for stripping comments from C header files, and interactive verification of the updating process
Plot	Computes and displays 3 dimensional functions in hires
Polygon	Moire type pattern generator with color cycling
QMouse	Queries whether a mouse button is pressed. This can give a return code that can customize a startup-sequence based on whether a mouse button was pressed.
Touch	Example of setting the datestamp on a file, using a technique from Commodore-Amiga
Trees	More extensive version of the trees program on Disk 31

Fred Fish Disk 50

Asm	Version 1.1 of a shareware 68000 macro assembler, compatible with the Metacomco assembler. This includes an example startup module and more Motorola mneumonics.
BreakOut	A brick breakout game, uses 3-D glasses
DiskZap	Version 1.1 of a program to edit disks and binary files
FirstSilicon	A smart CLI replacement with full editing and recall of previous commands
Missile	A Missile Command-type game, with sound, in assembler
PerfectSound	Sound editor for a low-cost sound digitizer
Sizzlers	Graphics demos
UnixArc	Ver of 'arc' for Unix System V machines,in C
Wombat	Version 3.01 of Dave Warker's terminal emulator

Fred Fish Disk 51

Bison	GNU for Unix 'yacc', working update to FF4
Compress	Update to the file compression program on Disk 6
Cos	"Wheel of Fortune"-type game in AmigaBasic
DifSeed	Unix-like 'dif' and 'seed' for finding the differences between two files, and then recreating the other, given one file, and the list of differences.
Sq, Usq	Portable versions of the CP/M squeeze and unsqueeze

Fred Fish Disk 52

Assign	Replacement for AmigaDOS 'assign' command in C
Fractal	Makes random fractal terrains
Poly, HAMPoly	Workbench-type demos for making polygons in lores and HAM
MxGads	Example of mutual exclusion gadgets with GadgetText
Tek4010	Tektronix 4010 terminal emulator
VDraw	Versions 1.16 and 1.19 of a Deluxe Paint-like drawing program

Fred Fish Disk 53

Animations	Demo animations with player program for Aegis Animator
ARCre	Creates rename scripts for files with long names, so they can be easily 'arc'ed and un'arc'ed.
ARP	Preliminary AmigaDOS replacements for 'break', 'cd', 'chmod', 'echo', 'filenote' and ' makedir'
Compiler	Not fully ported to the Amiga, this is a 68000 C compiler. It will produce simple assembly language output, but needs a lot of work.
Spreadsheet	Update with source of the 'vc' spreadsheet on disk 36
TarSplit	Port of program to split Unix 'tar'archives
UUencode	Utilities to encode and decode binary files for ASCII transmission, expanding them by 35%

Fred Fish Disk 54

Hanoi	Solves Towers of Hanoi Problem in it's own Workbench window. by Ali Ozer
ISpell	Port of a Unix screen oriented, interactive spelling checker. (Expansion RAM required) by Pace Willisson
Ing	A Screen of lots of bouncing little windows by Leo 'Bols Ewhac' Schwab
Lav	Displays number of tasks in run queue, averaged over last 1, 5, and 15 minute periods. by Willaim Rucklidge
MIDITools	Programs to play/record through the MIDI I/F. by Fred Cassirer
MoreRows	Program to make the Work Bench Screen larger than normal. by Neil Katin and Jim Mackraz
Tilt	Program to make your Amiga look like it didn't pass vibration testing. by Leo 'Bols Ewhac' Schwab

Fred Fish Disk 55

Csh	V2.05 of Matt Dillon's csh like shell (Modified for Manx C). by Matt Dillon, Modified by Steve Drew
NewStartups	New C Startup modules:
AStartup.asm	with 1.2 fixes and better quote handling.
TWStartup.asm	opens a stdio window, using user specs. by Commodore, posted to BIX by Carolyn Schepper
Palette	Change another program's screen colors. by Carolyn Schepper
PipeDevice	Allows the standard output of one process to be fed to the standard input of another. by Matt Dillon
ScreenSave	Save a normal or HAM mode screen as an IFF file. by Carolyn Schepper
ShanghaiDemo	Demo of the Activision game Shanghai.
SoundExample	A double buffered sound example for Manx C. by Jim Goodnow
Vsprites	A working vsprite example. by Eric Cotton
Vt100	V2.6 of Dave's Vt100 terminal emulator with kermit and xmodem. by Dave Wecker

Fred Fish Disk 56

ClipBoard	Clipboard device interface routines, to provide a standard interface. by Andy Finkle
ConPackets	Demos the use of DOS Packets, ConUnit, etc. by Carolyn Schepper
GetDisks	Program to find all available disk device names and return them as an exec list. by Philip Lindsay
GetVolume	Program to get volume name of the volume that a given file resides on. by Chuck McManis
Icon2C	Reads an icon file and writes out a fragment of C code with the icon data structures. by Carolyn Schepper
MergeMem	Program to merge the MemList entries of sequentially configured RAM boards. by Carolyn Schepper
mCAD	An object oriented drawing program, V1.1 by Tim Mooney

Fred Fish Disk 57

Replaced by FF97 Due to Copyright problems

Fred Fish Disk 58

ASDG-rrd	Extremely useful shareware recoverable ram disk. by Perry Kivolowitz
BigView	Displays any IFF picture, independent of the physical display size, using hardware scroll. by John Hodgson
EGraph	Reads pairs of x and y value from a list of files and draws a formatted graph. by Laurenece Turner
HyperBase	Shareware data management system. V1.5
MemClear	Walks through the free memory lists, zeroing free memory along the way. by John Hodgson
NewZAP	A third-generation multi-purpose file sector editing utility. V3.0 by John Hodgeson
RainBow	A Maurauder-Style rainbow generator. by John Hodgson
SMUSPlayers	Two SMUS plays, to play SMUS IFF music formatted files. by John Hodgson
View	A tiny ILBM viewer by John Hodgson
WBdump	JX-80 optimized workbench printer that does not use DumpRPort. by J. Hodgson

Fed Fish Disk 59

Browser	Update to browser program on disks 18 and 34. S-E
Browser2	Another different browser program. E
Clock	Clock program with fonts, colors. E
Dme	Dillon text editor V1.22 for programmers,ED
DropCloth	Puts pattern on Workbench backdrop,ED
DropShadow	Puts shadows on Workbench windows,E-D
FixWB	Similar to DropCloth, but doesn't work yet. S-D
mCAD	Object-oriented drawing program, version 1.2.2. Much improved over disk 56.
Robotroff	Demo of animated pointers on Workbench. S-E-D
Supermort	General compounding/amortization loan calculator. E-D

Fred Fish Disk 60

Various shareware and freeware programs

Blitz	Memory resident file viewer. Very fast. E-D
BlitzFonts	Makes text output faster. E-D
HandShake	Terminal emulator with VT52/VT100/ VT102support. E-D
Med	Mouse-driven text editor version 2.1. E-D
PrtDrvGen	Generates printer drivers, version 1.1.S available from author. E-D
Show	Slideshow-like IFF viewer, V2.1. E-D
Uedit	Customizable text editor V2.0. E-D
Ueturbo	Example Uedit setup macros. S-E-D

Fred Fish Disk 61

ATPatch	Patches Transformer to work under AmigaDOS 1.2. S-E-D
FillDisk	Writes zeroes to free blocks on a disk for security. S-E-D
LPatch	Patch for programs that abort when loading under AmigaDOS 1.2. S-E-D
MicroEmacs	Conroy MicroEmacs V3.8b, newer than disk 22. S-E-D
PearlFont	Like Topaz, but rounded edges.
Terrain	Generates fractal scenery. S-E-D
VSprites	Makes 26 Vsprites, from [illegible]

Fred Fish Disk 62

This is a port of the Unix game 'Hack', by the Software Distillery, version 1.0.3D.

Fred Fish Disk 63

This is a port of the Unix game 'Larn', by the Software Distillery, version 12.0B.

Fred Fish Disk 64

This is an offical IFF specification disk from Commodore, an update to disk 16.

Fred Fish Disk 65

Bawk	Unix text processor, like 'awk'. Doesn't work, but source is included. S-E-D.
MWB	Example of rerouting Workbench window open calls to another custom screen. Version 1.01, S-E-D
CloseWB	Example for closing a custom Workbench screen. S-E-D
Cookie	Generates one-line fortune-cookie aphorisms. S-E-D
JTime	Build-your-own mouse port clock.
MenuBuilder	Creates C source files for menus, based on text descriptions. S-E-D.
NewPackets	CBM tutorial on new packets and structures in AmigaDos 1.2.
PascalToC	Pascal to C translator, not so great. S-E-D
Prep	'ratfor'-like FORTRAN preprocessor. S-E-D
RunBack	Starts programs from CLI, allowing CLI window to close. E-D

SunMouse	This program automatically clicks in windows when the mouse is moved over them. V1.0, E-D

Fred Fish Disk 66

AmScsi	Preliminary plans for a SCSI disk controller board.
Asm68k	Macro assembler, version 1.0.1. E-D
Assigned	Example for avoiding DOS insert-disk requester, by scanning the list of 'assign'ed names. S-E-D
Dk	Pretends to eat away at CLI window. S-E-D
Flip	Flips whole screen as a joke. S-E-D
Foogol	Foogol cross-compiler generates VAX assembly code. S-E-D
Free	Prints amount of free space on all drives.S-E-D
MallocTest	malloc/free memory test program. S-E-D
Melt	Pretends to melt the screen. S-E-D
Nart	Graphic flying string demo. S-E-D
Purty	Easy way to set printer attributes from Workbench. E-D
RayTracer	Simple ray tracing program. E-D
SendPackets	Updated CBM examples of packet routines on disk 35. S-E-D
SnapShot	Memory resident screen dump. E-D
TagBBS	Shareware BBS system, version 1.02.

Fred Fish Disk 67

AmCat	Shareware disk cataloging program.
AmigaSpell	Shareware Intuition spelling checker, V2.0. E-D
Bouncer	3-D bouncing ball written in MultiForth, SED
Comm	Terminal program version 1.33, E
Dux5	Another version of DirUtil. S-E-D
HexCalc	Hex, octal, & decimal calculator. E-D
Icons	Various big and alternate image icons.
Mandala	Mandala graphics and sound. E
PersMait	Demo shareware personal file manager.
RSLClock	Menu bar clock version 1.3. E-D
RTCubes	Graphics demo of 3D cubes. E-D
Wheel	"Wheel of Fortune"-type game,AmigaBASIC

Fred Fish Disk 68

This is version MG 1b of the MicroGNUEmacs. Source and executable areincluded, as well as source for other computers besides the Amiga.

Fred Fish Disk 69

Asm68k	Macro assembler, v1.0.3, E-D
BlitLab	Blitter exploring program, in C, S-E-D
Conman	Replacement console device handler adds editing and history to any application that uses CON:, v0.9, E-D
Console	Replacement console routines, in C, S-E-D
Dk	Decays the screen bit by bit, update to disk 66, in Modula-2, S-E-D
Frags	Displays memory fragmentationby listing the size of free memory blocks, in C, S-E-D
IconType	Change the type of an icon, in C, S-E-D
Make	'make' in Manx C, S-E-D
MonProc	Monitors processes for packet activity, in C,S-ED
MouseClock	Mouse pointer into a digital clock,in C,SED
Sb	Browses system structures, from Transactor magazine, v1.0, in C, S-E-D
Spew	Generates 'National Enquirer'-type headlines from rules file. In C,S-E-D
Spool	Three programs to demonstrate multitasking & spooling in a printer spooler. In C, v1.2, S-E-D
Wc	Counts words ala Unix 'wc', but faster, in C,S-ED

Fred Fish Disk 70

This is a disk of shareware programs.

AmigaMonitor	Explores state of the system, v1.13
Arc	Standard file compressor and librarian, v0.23, a port of MS-DOS v5.0. E-D
BlackBook	Phone book program.
DoTil	Intuition-driven file manipulator program,v2.0.
GravityWars	Game of planets, ships and black holes,v1.03.
Jobs	Alternate user interface to CLI and WB, v2.1.
Lens	Magnifies area around mouse, shows it in a window, v1.0.
Life-3d	3D version of the classic cellular-automaton game, v1.2.
Logo	Logo language interpreter
SetKey	Demo keymap editor, v1.0
Vpg	Makes displays for aligning video monitors, v1.0.

Fred Fish Disk 71

AirFoil	Makes airfoils using the Joukowski transformation, in C, S-E-D
Amiga Basic	Miscellaneous programs including 3D plot program, a kaleidoscope, C-A logo drawing program file comparison utility string search program, S-E-D
Blocks	A variation of "lines", but with variable color blocks. E-D
Comm	Great terminal program, v1.34, E-D
DiskX	Utility for exploring file system.E-D
Fpic	Simple image processing program that operates on IFF pictures, with several filters, merging images, E-D
IconMk	Makes icons for files , v1.2a, E-D
Icons	New icons
NewFonts	Two new fonts; 'shalt18', an electronic ciruit element font, and 'ibm5', a PC-like font.
PetCLI	An AmigaBASIC CLI shell program.
PWDemo	Demo of the commercial product PowerWindows,v1.2. It aids creation of custom windows, menus, and gadgets, giving C or assembly source. E-D
Rot	Creates and animates 3-D objects, v0.5, E,D
TimeSet	Sets time from Workbench, E-D

Fred Fish Disk 72

This is a disk of IFF pictures.

Fred Fish Disk 73

Add	Customizes existing program menus with Amiga-key shortcuts. Also includes 'until', which waits until a given window is created. Shareware, in C, S-E-D.
AutoIconOpen	Fools WB into thinking mouse has double-clicked icons. In C, S-E-D
Dio	Generic Exec device interface code for opening libraries, getting multiple I/O channels, asynchronous operations, etc. In C, S-E-D.
Dissolve	Slowly displays IFF files, ala Nov 86 Dr. Dobb's program. In C, S-E-D
DTerm	Flexible, reprogrammable terminal program v1.10, E-D
Expose	Re-arranges windows so that at least one pixel of menu bar gadgets are exposed. InC, S-E-D.
Lit	Scans a text file, converts to C-style printable strings.C,v2.0, S-E-D
Lmv	"Long Movie", program views series of IFF picts in quick succession, upto 19 fps. Shareware, E-D
MouseOff	Mouse pointer disappears after ten seconds of non-use. In C, S-E-D
ParOut	Examples of controlling parallel port with resources instead of the PAR: device.In C,S-E-D
PenPalFont	Child-like font.
RunBackGround	Similar to RunBack on disk 66, runs program from the CLI allowing the CLI window to close. In C,S-E-D
SnapShot	Screendump utiliy,update FF 66.E-D
TypeAndTell	Example installs a device handler before Intuition, and speaks each key as it is pressed. In C and assembler, S-E-D
Xplor	Prints info about system lists, in assembler,S-E-D

Fred Fish Disk 74

Cled	Edits and recalls CLI commands, v1.3, E-D
Control	Intercepts graphic printer dump calls and accesses color map, width,and screen resolution. C,S-E-D
Dme	Simple WYSIWYG text editor for programmers,v1.25. Update of FF 59.E-D
DropShadow	WB dropshadows, v2.0.UpdateFF59. E-D
Funds	AmigaBASIC prog tracks mutual or stocks-D
Less	Text viewing program, like Unix "more", v1.1, update to disk 34. S-E-D
Makemake	Scans C source files and constructs a vanilla 'makefile' in the current directory. S-E-D
mCAD	Object-oriented drawing prog. v1.2.4, update to FF 59.Shareware, E-D
Random	Simple random number generator in C. S-E-D
TDebug	Monitors devices by intercepting Exec SendIO() and DoIO() vectors, in C, v1.0, S-E-D
Units	Converts measurements in different units, includes 'chart' option, in C, S-E-D
XCopy	Replacement for AmigaDOS 'copy', doesn't change the date, uses Unix wildcards. E-D

Fred Fish Disk 75

Bezier	Play with Bezier curves points and granularity, S-E-D
BSplines	Play with b-splines, as above, S-E-D
Comm	C source for Comm terminal program v1.34. S-E-D
Copy	Replacement 'copy' command v1.0, preserves date, in C, S-E-D
Diff	Simple 'diff' in C, S-E-D
DuM2	Another DirUtil in Modula-2, v1.5, S-E-D
Eless	Fast 'dir' program in C, S-E-D
Fd	Faster 'eless' in C, S-E-D
HardCopy	Sends a transcript of a CLI session to a file, in C, S-E-D
MouseOff	Update FF73, turns off mouse pointer, S-E-D
SetFont	Changes the font in a Workbench screen, v2.0, S-E-D
SpeedDir	Another fast 'dir', in assembler, S-E-D

Fred Fish Disk 76 & 77

These are disks 1 and 2 of Chris Gray's Draco distribution for the Amiga. Draco is a compiled, structured language reminiscent of both C and Pascal. A full interface to AmigaDOS and Intuition is supplied. Be sure to get both disk 76 and 77.

Fred Fish Disk 78

Cycles	Cycle game like "Tron", v1.0, E-D
EOMS	Experts Only Mercenary Simulator game, E-D
MandelVroom	Mandelbrot generator with enhanced palette controls, fixed/floating point, presets, v1.50, in Manx C, S-E-D

Fred Fish Disk 79

AsmTools	CLI tools in assembler: echo, loadit, mounted, setlace, why; S-E-D
AssignDev	Give devices multiple names, in C, S-E-D
AuxHandler	Example of a dos handler that allows use of a CLI via the serial port. Includes source. Author: Steve Drew
Cmd	Redirects printer output to a file, in C, S-E-D
Info	AmigaDOS 'info' replacement, in C and assembler, S-E-D
Kill	Removes a task and its resources, in C,S-E-D
M2Error	Displays errors from TDI Modula-2 compiles, S-E-D
MonProc	Update to process packet prog. from FF69,in C,S,E,D
Mounted	Program for testing if a drive is present, in a script In C, S-E-D
Nro	Another 'roff'-style text formatter, in C, S-E-D
ParTask	Finds parent task, in C, S-E-D
QueryAny	For scripts, asks a question, accepts Y/N, gives return code. In assembler, S-E-D
ScnSizer	Resets pref settings for screen size, in C,SED
SharedLib	Example, shared lib, in C & assembler, S-E-D
Task	Simple CreateTask() example in C, S-E-D
Uw	Unix Windows client v1.0, in C, S-E-D
Who	Lists tasks on ready and wait queues, in C, S-E-D

Fred Fish Disk 80 (see Fred Fish 90)

Fred Fish 80 has been withdrawn due to copyright problems.

Fred Fish Disk 81

Asm68k	V1.1.0 of a macro assembler
AutoFacc	Shrinks the FACC window and moves it to the back
Brushes	53 custom IFF brushes of electronic symbols
CheckIFF	Checks structure of an IFF file CledV1.4 update FF74 of a simple CLI
Conman	Replaces console handler to add editing and history to many programs
Fonts	Miscellaneous fonts
Icon	V6.0 of the Icon programming language
KeyLock	Freezes the keyboard and mouse until pass word entered.
ScatDisplay	hack created from "lng"
Smush	Smushes an IFF file.
Target	Each mouse cliko becomes a gunshot

Fred Fish Disk 82

Adventure	Port of the classic Crowther and Woods game
AmicTerm	V0.50 of a telecommunications program, with scripts, redial, beeps, enhanced file requester
D2D-Demo	Demo version of Disk-2-Disk by Central Coast Software
DX-Synth	Voice filer program for Yamaha DX series synthesizers, update to disk 38
DiskMan	V1.0 of another DirUtil program
Icons	Miscellaneous new icons
Panl	Universal MIDI patch panel, v1.2
Rocket	Another Workbench hack, plays Lunar Lander
Sand	Game of sands following your pointer.

Fred Fish Disk 83

This disk contains a demo version of TeX from N Squared. It is limited to small files, and the previewer can only display ten pages or less, and only a small number of fonts are provided.

Fred Fish Disk 84

AudioTools	Programs from Rob Peck's July/August Amiga World article
BlitLab	Blitter experimentation program, V1.2, update to FF69
Ed	Simple editor, similar to Unix 'ed', based on the editor in Software Tools.
GravityWars	Game of planets, ships and black holes, v1.04, update to disk 70.
HunkPad	Adds legal padding to executables for Xmodem transmission.
PipeHandler	An AmigaDOS pipe device which supports named pipes and taps. V1.2
PopCLI	V3.0 of a hot-key to invoke a CLI window, with screen blanker, update to disk 40.
Requester	Update FF34, file requester similar to DPaint.
ScottDevice	V33.1 of a 'mount'able MicroForge SCSI driver.
Viacom	Another Schwab hack, makes TV-like static on screen.Parody

Fred Fish Disk 85

Csh	V2.06 of Dillon's 'csh'-like shell
FileReq	Source to wildcard file requester
Hide	Hides expansion memory from programs
ImageTools	Shareware tools to manipulation IFF images
LowMem	ServerShared library to aid in low memory situations
Plot5	A star plotting program with source.
RawIO	Example of setting raw mode on standard input
Rocket	Lunary Lander for Workbench, with source.
VMore	'more'-like text viewing utility, v1.0 SE
Vnews	Simple Unix news reader.

Fred Fish Disk 86

AutoPoint	Auto-selects window under the mouse pointer, with screensaver.
ClickToFront	Double-clicks in window brings it to front, v1.1, S-E-D
Cmd	V3.0 of a tool to redirect printer output toa file.
FileIISG-Demo	Demo of Softwood File IIsg, a database manager with sound and graphics.

Fred Fish Disk 87

AdvSys	Adventure system from Byte May 1987, v1.2 E-D
AutoIconOpen	Fools Workbench to open disk icons, V1.2 update to disk 73, S-E-D
Claz	Converts IFF files to PostScript, V2.0, SED
Commodi	tiesMackraz's Commodities Exchange, an exec library to manage input handler, v0.4
Diff	Update to disk 75 of Unix-like 'diff', S-E-D
Dme	V1.27 of Dillon's text editor, update FF74,E-D
DropShadow	V2.0 of prog. that puts shadows on Workbench, S-E-D
Elib	Shared library example in Manx C.
ID-Handler	An AmigaDOS device handler generates unique identifiers, V1.0, S-E-D
Install	Alternate AmigaDOS 'install' programs,SED
MemWatch	Waits for low memory trashing, V2.0, SED
MovePointer	Moves pointer to given location, S-E-D
MoveWindow	Move window to given location, S-E-D
MunchingSq	Munching Squares hack, S-E-D
PalTest	Test to see if this is a PAL machine, S-E-D
Sc	Generates random scenery, S-E-D
Tek4695	Tek4695 printer driver
WBDualPF	Example of dual-playfield screen, update FF41, S-E-D
WarpText	Fast text rendering routines, S-E-D
Yaiff	rExample IFF reader, S-E-D
Zoo	A file archiver like "arc", v1.42A, E-D

Fred Fish Disk 88 (see Fred Fish 89)

FF Disk 88 has been removed due to copyright problems

Fred Fish Disk 89 (replaces Fred Fish 80)

DirMaster	Disk catalogue program, V1.0a, E-D
FuncKey	Shareware function key editor, V1.01, E-D
MFF-Demo	Demo of MicroFiche Filer database prog
ScreenShift	Adjust screen position ie Preferences,SED
Snake	Bouncing squiggly lines demo, S-E-D
AutoEnguirer	screen contraption requester improvement S-E-D
DemoLition	Display Hack S-E-D

Fred Fish Disk 90 (replaces Fred Fish 80)

AmiGazer	Night sky viewer of 1573 stars, set date, time, day. E-D
CardFile	AmigaBasic card file study aid. E-D
Conman	Console handler replacement gives line editing and history to most progs, v0.98,ED
MandelVroom	Slight update to disk 78 Mandelbrot program, E-D
NewDemos	Replacements for lines and boxes demos that take less CPU time, E-D
Othello	Game of Othello, E-D
PrinText	Displays text files with gadgets, speech, IFF display, v1.2, E-D
PrDrvGen	Automatic printer driv. generator,v2.2b,ED
RainBench	Cycles colors of WB backdrop or text. ED
ShortCut	Makes single-key shortcuts for entering commonly typed CLI commands,&custom macros. E-D
ShowPrint	Displays and prints all sizes of IFF pictures & controls printer output styles, v2.0 E-D
Sizzlers	Graphics demos, v1.7.0, E-D
Timer	Small Workbench timer counts time and $/ minute, E-D
Tools	Inovatronics tools: a memory editor, memory disassembler, ASCII chart, and calculator. E

Fred Fish Disk 91

Adventure Definition Language (ADL) a superset of an older language called DDL by Michael Urban, Chris Kostanick, Michael Stein, Bruce Adler, and Warren Usui. ADL enhancements by Ross Cunniff. Included are sources to the ADL compiler, interpreter, and debugger. Binaries combined by Ross with Lattice 3.03. CLI enviornment only. Documentation is available from the authors.

Fred Fish Disk 92

As6502	portable 6502 assembler, C source, by J. Van Ornum, Amiga port by Joel Swank
Bawk	Text processor update from FF65 Inspired by UNIX awk. Searches files for patterns, performs actions based on patterns. By Bob Brodt; Amiga port by Johan Widen
HunkPad	update of FF84 version. by J. Hamilton,pads an object file to a multiple of 128 bytes for better xmodem transfer. S E
Less	Like Unix "more", better, version 1.2 update of FF74. Scrolls Back and forward. S E by Mark Nudelman, Amiga port by Bob Leivian.
Ndir	Library that implements the 4BSD unix dir access routines by MikeMeyer. S
Parse	Recursive descent expression parcer, computes, and prints expressions. includes transcendental function support. c Source included. by J Olsen
Shar	Two programs to pack and unpack shell archives includes C source. by Fabbian G. Dufoe
SmallLib	8 times smaller Amiga.lib replacement, binary only. by Bryce Nesbitt
UUencode	Encode/decode binary files for e mail or text-only methods. Update of FF53, includes checksum technique, compatible with older versions, plus transparent to older versions options. By Mark Horton, modified by Alan Rosenthal and Bryce Nesbitt.

Fred Fish Disk 93

Dme	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, title-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code. by Matt Dillon
MicroEmacs	Version 3.8i, update to FF61 includes source. Orig by Dave Conroy multiple modifications by Daniel Lawrence

Fred Fish Disk 94

AudioTools	Demo programs from Rob Peck's July/Augustt issue of AmigaWorld on accessing the audio device. V2,update of FF84. S. by Rob Peck
ClickUpFront	Similar in function to ClickToFront prog (FF86), bring windows tofront by clicking on any part of them. V 1.0. by Davide Cervone SE
HeliosMouse	Automatically activate a window simply by moving the mouse pointer into the window. V 1.0. Includes source. By Davide Cervone
IFF2Ps	Convert any IFF file to postscript for printing or viewing on a postscript compatible device. Version 1.2, by William Mason and Sam Paolucci E
ModulaTools	Various Modula 2 programming routines. by Jerry Mack
Terrain3d	Pseudo-random 3d relief scenery generator, update of "sc", FF87. by Chris Gray, 3d by Howard Hull

Fred Fish Disk 95

Cmd	redirects the serial.device or parallel. device output to a file. Capture print jobs, debug or "offline" printing.V4 By C Scheppner SE
CygnusEdDemo	Demo of CygnusSoft's CygnusEd editor, a multiplefile, multiple feature editor.Includes demo 3.0 of MandFXP. by CygnusSoft Software E
Gomf	"Get Outa My Face" makes the Guru go away to allow clean-up & shutdown more cleanly. V 1.0, by Christian Johnsen E
Journal	records sequence of mouse & keyboard events, stored in a file for future playback. Good for demos or documenting bugs. E. by D. Cervone
MergeMem	attempts merging of MemList entries of sequentially configured ram boards. When successful, allows allocating a section of memory which spans both boards. V 2, update of FF56. by Carolyn Scheppner SE
PrinterStealer	Asimilar to "Cmd", allows diversion of output destined for printer to a file. Binary only, Source avai. from authors. by A. Livshits & J-M Forgeas
Record-Replay	similar to "Journal", records and plays back mouse and keyboard events. B only. source avail. from authors, Alex Livshits & J-M Forgeas

Fred Fish Disk 96

AnimPlayer	Animation reader and displayer by the combined efforts of Videoscape, Sculpt3D, Silver, Forms- In-Flight, and Animator:Appretiocby M Hashetal.
Chess	Amiga port, non-Amiga interface. High playability. V 1.0, S. by J. Stanback, Amiga port by B. Leivian
Hackbench	provides source for WB-Likeprog, for experimentation & validation of new interface ideas. Not a WB replacement. by Bill Kinnersley
Label	Print labels with arbitray text. V 1.3, Source available from author, M.Hansen
LineDrawer	Produces line drawings based on drawing commands stored in a text file. Includes demo that draws an outline map of the USA and state borders. V1.0, SE. by John Olsen
PopUpMenu	Example code implementing pop-up menus, reasonably compatible with Intuition menus.SE. by Derek Zahn
Tek4695	Tektronix 4695/4696 printer driver. SE. by P Staub
TimeRam	Fast and Chip ram test prog.E by B Takahashi
WarpText	Fast text rendering routines, to be linked with application progs.Text display "as fast or faster than "blitz". V2.0 update of FF87, S by Bill Kelly

Fred Fish Disk 97

Replaces FF57 for Copywrite problems

CutAndPaste	Implementaions of Unix cut and paste commands. by John Weald

GraphIt	Progran to plot simple functions in 2 or 3 dimensions. by Flynn Fishman
Juggler	V1.2 of robot juggler animation. Uses HAM mode and ray tracing. by Eric Graham
MouseReader	Shareware program to read text files & view IFF files using only the mouse. by William Betz
Splines	Prog to demonstrate curve fitting & ren-dering techniques. by Helene (Lee) Taran
Shm	Simple graphics demo, approximately simulates the motion of two interacting pendulums. Includes Source by Chris Ediss

Fred Fish Disk 98

Access	16 color terminal program based on Comm V1.,34. Includes Macro window, custom gadgets, colorized menus, etc. V. Beta 0.18 by Keith Young ,comm by D.J.James. E.
Backup	Writes AmigaDos disks as the backup destin-ation. recover files from the backup disk. Requires manual decisions on disk structure. by Alan Kent SE
DCDemo	DishCat 2.3, a disk catalog program, demo limited to cataloging 100 files at a time. by Ed Alford, MicroAce Software
HdDriver	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED
QBase	Quick-Base, a "MailBase Management utility", define and maintain a maximum of 200 records per file. by Kevin Harrise E
Thai	Thai language quiz program. Speak or type english/Thai sentences from supplied file. by Alan Kent SE

Fred Fish Disk 99

A-Render	Version .3 a Ray- Tracing Construction Set for the Amiga Computer by Brian Reed ED

Fred Fish Disk 100

Berserk	Must see animation, by Leo Schwab
Conman	Console handler replacement, provides line editing and command line histories transparent to application prog uses CON: windows. Shareware V1.0 by W Hawes. E.
WBLander	Workbench display hack game, upgrade of "Rocket" on FF85, now with sound effects. By Peter da Silva. E

Fred Fish Disk 101

CirPlane	Circular plane generator for VideoScape3D. Generates a clockwise circular polygon with the specified number of vertibles. V1.0 by T Floryan SE
IconAssembler	Change Workbench Icons with IFF-brush files by Stefan Lindahl E
Microspell	Standalone spelling checker scans text files and reports errors. 1000 common word list, 43,000 word main dictionary with multiple user dictionary support. Interfaces with MicroEMACS 3.9 with an emacs macro to step through the source file, stopping at suspect words and allowing the user to option. V1.0 by Daniel Lawrence, SED
Midi	midi library and utility set. Includes Midi monitor, routing utility, status utility, and more. by Bill Barton SED
PsIntrp	Postscript Interpreter reads and previews files on screen. by Greg Lee S(assy)E
StartUps	Three C startup file replacements for standard Astartup.obj and LStarup.obj. Options include (1) BothStartup.obj, for the WorkBench programs or CLI programs with or without command line parameters. (2) WBStartup.obj, for WorkBench programs or CLI programs that require no command line parameters. (3) CLIStartup.obj for CLI programs that require command line parameters but do not need to be WorkBench runnable. by Bryce Nesbitt SE

Fred Fish Disk 102

Dbug	Machine independent macro based C de-bugging package. Update FF41. by F Fish profiling support by Binayak Banerjee SE
Match-stuff	Heavy duty text pattern matching stuff. includes simple match text replacement capability. By Pete Goodeve
Sectorama	Recover lost or damaged data from floppy or hard disks or repair a damaged volume. by David Joiner E
SiliCon	Smart input line interpreter with window for full editing. Upgrade FF50 by P Goodeve., E
Xicon	Use icons to call up scripts containing CLI commands. V2.0 upgrade of FF31.by Pete Goodeve E

Fred Fish Disk 103

AviTrees	Library and test prog. implement routines for creating and using trees held in memory.S.
Calc	A programmable RPN calculator.
Cref	A C cross ref. prog. S.
DosKwik	A pair of progs. allows you to save files to one or more floppies for quick loading. Doesn't store Dos format.
IntuiDos	A prog. to improve control and handling of the material on all disks in 'CLI-area'.
MFF-Update	A text import util. for MicroFiche Filer (demo on FF 89) and updates to some PD disk library databases.
Pack-It	Takes all files the files and dirs. on a disk & packs them into a single file. for modem.
Sol	Amiga version of solitaire.

Fred Fish Disk 104

Analyticalc	Is a large and powerful spreadsheet prog.

Fred Fish Disk 105

AsmProgs	Misc. assembly tools. Includes some S.
BasicProgs	LeastSquare solves least square probs & graphs results. S.
Bison	A replacement for unix "yacc" command. S.
Dmouse	Another prog in the tradition of display hacks". S.
FlamKey	Allows keyboard and mouse inputs to be locked until a password is entered.
GravityWars	Game of planets,ships and black holes, v2.0 update to FF84.
IPo2C	A util. to write a C-lang definition to mimic the intuition pointer.S
Pere-et-Fil	Ex. of creating & using reentrant processes. S.
Record Replay	Similiar to "Journal" v2.0 update to FF95.

Fred Fish Disk 106

Funckey	Shareware function key editor, v1.1update to FF89. Source avail. from author(Anson Mah).
MoreArt	A small selection of some Amiga artwork.
QuickFlix	An IFF slideshow and cel animation prog,v0.13.
RistiNolla	A Finnish game. Also called Go-Moku. v1.0

Fred Fish Disk 107

Csh	V2.07 of Matt Dillon's csh like shell.S.
Diff	A util.,similiar to other common "diff" programs.S.
ProSuite	Suite provides ex. code of facilities such as FileIO Requester, XText, DoRequest. & tutorial on how to program the Amiga. Book 1.01.S
SVTools	Some useful tools. S.

Fred Fish Disk 108

AList	Dir listing prog. based on LD4 prg S
DirMaster	Disk cataloger. v1.0b, update to FF89. S.
Dots-Perfect	Printer Driver for an Epson MX80 printer with upgrade kit installed. S.
MonIDCMP	Lets you monitor the IntuiMessages that pass through an IDCMP window. Prints the message class,mouse coordinates,qualifier values. Great for debugging. S.
PrintPop	A util. to send common control settings to PRT: device. S.
Sectorama	Utilities to recover lost or damaged data from floppies & hard disks. v1.1, an update toFF102.
Tek	Vt100 emulator for a Tektronix 4010/4014. (V2.6) update to FF52. S.
Zoo	File archiver, like "arc". v1.24B. update to FF87

Fred Fish Disk 109

Machine	A new animation.
SimCPM	A CP/M sim,simulates 8080 along with h19 emulation. S.
UUpc	Hook up your Amiga as a usenet node. S.

Fred Fish Disk 110

A68k	A 68000 assembler written in C. S.
Pdc	An optimizing C compiler for the 68000 processor. update to FF53. but not based on the code of that disk.

Fred Fish Disk 111

AmyLoad	A graphical monitor of cpu, blitter, & memory use. Includes two components; load.device,monitors system parameters, & amyload, which is the user interface & display program. by Jeff Kelley SE
AssignDev	Assigns multiple names to a given device. modified version of the original released on disk number 79. By Phillip Lindsay, mod by Olaf Seibert SE
Gauge	Continuously displays memory usage in a vertical bar graph Binary only. By:Peter da Silva
HeliosMouse	Another "sunmouse"prog. Automatically activates a window by mouse pointer V 1.1, update to disk 94. By:Davide Cervone SE
Labels	Alphabetic & numeric ordered cross reference lists of defined system constants. Recommended for debugging purposes only, use the symbolic values in progs! By:Olaf Seibert
Mandel	Another mandelbrot generator program, with bits & pieces of code from C. Heath & R.J. Mical. By:Olaf Seibert S
PopLife	A PopCLI type that plays life all over your screen. Lots of bits & pieces from Tomas Rokicki's blitlab & John Toebes' PopCLI. By:Olaf Seibert S

Fred Fish Disk 112

BeachBirds	Beach scene portrayed by sprites & sound 512K machine. By:Jerrold Tunnell B only.
Bully	Pushes all open screens around (thus the name "bully"). Show more than one demo at a time By:Mike Meyer S
DropShadow	Dropshadow V2.0, use with Bryce Nesbitt's Wavebench demo. B only. By:Jim Mackraz
HagenDemos	"RGB" & "Focus". RGB requires one meg. B only. By:Joel Hagen
Viacom	Latest version of viacom for use in conjuction with WaveBench demo. B only. By:Leo Schwab & Bryce Nesbitt
WaveBench	A neat screen hack, & runs on 512K machines. For more laughs, try in conjunction with Viacom or Ds (Dropshadow). Includes S. By:Bryce Nesbitt

Fred Fish Disk 113

AmiCron	Simple Unix "cron" type program,a background task uses a disk-resident table to automatically run certain tasks on a regular basis, at specific times. V 2.3, includes S. By:Steve Sampson, Amiga port by Rick Schaeffer
Dme	V 1.28f of Matt's text editor. A simple WYSIWYG editor designed for programmers. It is not a WYSIWYG word processor Features include arbitrary key mapping, fast scrolling, title-line statistics multiple windows, & ability to iconify windows. Update to FF93, includes S. By:Matt Dillon
DosDev	Example DOS device driver in Manx C. Version 1.10, includes S. By:Matt Dillon
M2Amiga	Demo of the final product M2Amiga. A fast single pass Modula-2 compiler with editor, linker, a small set of interface & standard libraries. Compiles only small demo programs by limiting codesize & imports. Further development of the ETHZ compiler on Disk 24. B only. Demos with Source. By:R. Degen, C. Nieder, M. Schaub, J. Straube (AMSoft)
NoIconPos	Clears position info of any icons, allows WorkBench to pick a new place for the icon. Useful for disk & drawer icons where Snapshot rewrites the icon & the window information. Modula-2, another demo for M2Amiga By:Markus. Schaub

Fred Fish Disk 114

CDecl	English to C (and vice versa) translator for C declarations, a must for anyone except possibly the most hardcore C guru. By:Graham Ross, S
Vt100	V2.7 of vt100 terminal emulator with kermit & xmodem file transfer. Includes a few bug fixes posted to Usenet shortly after the posting of v2.7. Update to FF55. Includes S. By:Dave Wecker
WBLander	a special version of the WBLander program from FF100. Ending is unique. Effective use of sound, Includes S. By:Peter da Silva & Karl Lehenbauer

Fred Fish Disk 115

Killer	Masterful Video commercial of the Amiga, Beatles music, requires one meg of memory to run. Binary only. By:R. Wilt
Marketroid	Another devious sprite oriented demo with lots of "in" jokes. 512K required, includes S. By:Leo Schwab

Fred Fish Disk 116

Movies	A ram animation system with three different example animations; Kahnankas, Rocker, & F-15. Kahnankas & Rocker run on a 512K Amiga & show off overscan HAM mode. Includes a animation player program (movie), animation builder programs (dilbm, pilbm), & a text/graphics display program (vilbm). By:Eric Graham & Ken Offer

Fred Fish Disk 117

AMUC_Demo	A really neat horizontal scrolling demo that is a 2400 x 200 pixel 32 color IFF picture composed of digiview snapshots of members of the AMiga Users of Calgary, superimposed on a very wide picture of the Calgary Skyline. B only. By:Stephen Vermeulen & Stephen Jeans
ExP_Demo	Demo version of Express Paint 1.1., used to create the scrolling demo picture in the AMUC_Demo drawer on thi disk. B only. By:Stephen Vermeulen

Fred Fish Disk 118

Empire	This is a complete rewrite, from the ground up, in Draco, of Peter Langston's Empire game. A multiplayer game of exploration, economics, war, etc, can last months.Played either by local keyboard or through modem.V1.0, shareware, & includes S code. By:Chris Gray, original game by Peter Langston
HAMmmm	Displays lines whose end points are bouncing around the screen, which is a double buffered HAM screen. The Y positions of the points are continuously copied into an audio waveform that is played on all four channels, & the pitch of a just intoned chord is derived from the average X position of these points. JForth. Source By:Phil Burk
Stars	Based on original code by Leo Schwab, has credits longer than the actual demo. Runs on 512K Amiga. B only. By:Hobie Orris
WireDemo	Demonstrates the Amiga's line drawing speed. Runs on a 512K Amiga. Includes S. By:Matt Dillon

Fred Fish Disk 119

MicroEMACS	Version 3.9e of Daniel Lawrence's variant of Dave Conroy's microemacs. This is an update to the version released on disk 93. Also included, for the first time, is extensive documentation in machine readable form. Includes source. Author: Dave Conroy, MANY enhancements by Daniel Lawrence

Fred Fish Disk 120

Amoeba	This clone of Space Invaders is one of the best freely redistributable games for the Amiga to date. Unlike many commercial games, it even works correctly in a multitasking environment (by not requiring you to reboot just to play a game). Highly recommended! Binary only. Author: LateNight Developments
BackGammon	A graphical Backgammon game done as an undergraduate A.I. course project. Version 1.0, includes source. Author: Robert Pfister
Bankn	A complete checkbook system offered by the author as shareware. Version 1.3, binary only. Author: Hal Carter
EgyptianRun	Cute little "road race + hazards" type game. Version 1.1, binary only, shareware, source available from author. Author: Chris Hames
IconImage	Program to replace an old icon image with a new image, without affecting icontype, drawer data, etc. Includes source. Author: Denis Green

Fred Fish Disk 121

BasicStrip	An AmigaBASIC program that helps to convert programs written in other forms of Basic to AmigaBASIC. Author: George Trepal
DataPlot	A shareware plotting program written in AmigaBASIC. Also includes a least squares curve fit program. Author: Dale Holt
Plot	A shareware 3-D graphing program written in AmigaBASIC, with some sample output plots. Source available from author. Author: George Trepal
Stairs	This AmigaBASIC program demonstrates a musical illusion based upon perceptual circularity of widely spaced tones whose volumes are defined as a sinusoidal relationship to their frequency. Author: Gary Cuba
Uedit	Version 2.3 of this nice shareware editor. Has learn mode, a command language, menu customization, and other user configurability and customizability features. Binary only, shareware, update to version on disk 60. Author: Rick Stiles
WBColors	A simple little program to change the Workbench colors to a predetermined color set, for programs that expect to be booted off their distribution disk but instead are run from a hard disk. Includes source. Author: Stefan Lindahl

Fred Fish Disk 122

Asteriods	Nicely done "blast the asteriods" type game. Unique feature is that all the images and sounds are replaceable by the end user. So instead of ships and rocks, you can have an Amiga against a horde of IBMPC's if you wish. Author: Rico Mariani
Iff2Pcs	An interactive puzzle program that takes any IFF file containing up to 16 colors, and breaks it up into squares to make a puzzle which the user can then piece back together again. Version 1.0, includes source. Author: Ali Ozer
Names	A shareware program to create and manage mailing lists. Binary only. Author: Ernie Nelson
Pr	A little utility to print listings in different formats. Similar to the Unix "pr" program. Includes source. Author: Samuel Paolucci
PushOver	A neat little board strategy game, in AmigaBASIC. Push your pieces onto the board until you get five in a row in any direction. Includes source. Author: R.Yost
PuzzlePro	Create a puzzle from an IFF picture, which the user can then piece back together again. Written in AmigaBASIC. Version 1.0, binary only, shareware, source available from author. Author: Syd Bolton

Fred Fish Disk 123

Arp	ARP stands for "AmigaDOS Replacement Project". Arp is an effort led by Charlie Heath of Microsmiths Inc., to replace the current DOS in a compatible fashion, so that current programs will continue to work. Arp also makes whatever improvements are possible, so that current and future programs will work better. Author: Various authors contributed work
Car	This animation is one of Allen's entries to the Badge Killer Demo Contest. It apparently is an inside joke relating to a well known Amigan's experience with a certain highend graphics hardware manufacturer. Author: Allen Hastings

Fred Fish Disk 124

Icons	Some sample animated icons. You might find just the icon for that refugee CLI program you've been meaning to make runable from the WorkBench environment. Author: L. Pfost
Tarot	An AmigaBASIC program written by the author as an exercise for learning BASIC. Contains some nice graphic renditions of tarot cards. Author: L.Pfost

Fred Fish Disk 125

ElGato	This animation is Kevin's entry to the Badge Killer Demo Contest. It also has a background music arrangement, requires Sonix to use. By Kevin Sullivans

Fred Fish Disk 126

Colour	A program to manipulate the colors of specific named screens, saving their current color sets to data files, loading new color sets from data files, or interactively changing the colors. Includes source. By J. Russell
Dance	These two programs, "dancing polygons", are John's entry to the Badge Killer Demo Contest. They are variations of one another, but demonstrate the range of colors available on the Amiga. Includes source. Author: John Olsen
HBHill	This animation is one of Kevin's entries to the Badge Killer Demo Contest. It is the first known animation that makes use of the Amiga's "Extra Half Brite" mode. Author: Kevin Sullivan
Iconify	A subroutine that creates an icon on the Amiga screen that can be subsequently dragged around, and double-clicked on. You can use this to have your programs "iconify" themselves to temporarily get out of the user's way. With source and demo program. By Leo Schwab
OnlyAmiga	This animation is Iqbal's entry to the Badge Killer Demo Contest. It consists of tree balls being juggled by pyramids rotating on their tops. By Iqbal Singh Hans
Suplib	The support library needed to rebuild various programs of Matt's from the source, including DME, DTERM, etc. Includes source. Author: Matt Dillon
VCheck	Version 1.2 of the virus detection program from Commodore Amiga Technical Support. This version will test for the presence of a virus in memory, or on specific disks. Binary only. Author: Bill Koester.

Fred Fish Disk 127

Bounce	This program is Steve and Tom's entry for the Badge Killer Demo Contest. It creates little dots that bounce around and multiply. Includes source. Author: Steve Hansel and Tom Hansel
Nemesis	This demo is Mark's entry to the Badge Killer Demo Contest. It is quite small for what it does, and won fifth place in the contest. Binary only. By Mark Riley
Ripples	This animation is one of Allen Hastings' entries to the Badge Killer Demo Contest. Unlike most other animations, it shows a fixed object from a moving point of view, rather than a moving object from a fixed point of view. Author: Allen Hastings

Fred Fish Disk 128

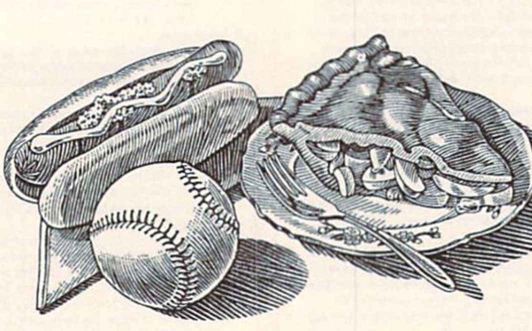
Dis	A 68000 disassembler, written in 68000 assembler. Includes source. Author: Greg Lee
DropCloth	lets you place a pattern, a 2 bitplane IFF image or a combination of a pattern and image, into the WorkBench backdrop. Version 2.2, shareware, binary only. Author: Eric Lavitsky
LedClock	An extremely simple clock program, for interlaced screens only. Includes source. Author: Ali Ozer
MRBackUp	A hard disk backup utility, that does a file by file copy to standard AmigaDOS floppy disks. Includes an intuition interface and file compression. Version 1.3, includes source. Author: Mark Rinfret
Paint	A simple screen painting program, written in web. Requires web preprocessing program to rebuild from source. Includes source in web. Author: Greg Lee
PrtDriver	A printer driver for the Toshiba "3 in one" printer in its Qume (best) mode. Includes source in C and assembler. Author: Rico Mariani
SDBackUp	A hard disk backup utility. CLI interface only. Does file compression. Version 1.1, binary only. By Steve Drew
Sed	A clone of the Unix sed (Stream EDitor) program. Includes source. Author: Eric Raymondw
Keys	A "hot-keys" program that binds keyboard function keys to window manipulation functions (window activation, front to back, moving screens, etc). Includes source. Author: Davide Cervone

Fred Fish Disk 129

DosKwik	A pair of programs which allow you to save files, or a group of files, to one or more floppies for quick loading, does not store files in DOS format, which is why it is faster. V2.0, update to FF103. Binary, Shareware. Author: Gary Kemper
MRBackUp	A hard disk backup utility, does a file by file copy to standard AmigaDOS floppy disks. Includes intuition interface & file compression. Versions 2.0 (with sources) and 2.1 (binary only, source available from author). Update of FF128. By Mark Rinfret
PaintJet	HP PaintJet printer driver from HP sources.
Patch	Two independent ports of Unix utility "patch", which applies context diffs to text file to automatically update them. Patch version 1.3 was ported to the Amiga by Rick Coupland and patch version 2.0 was ported by Johan Widen. Includes source. Author: Larry Wall

Fred Fish Disk 130

DirMaster	Shareware disk cataloger, V1.1, update of FF108, new features and enhancements. Binary only. By Greg Peters
Evo	Human evolution toy/tutorial with source.By S. Bonner
Hp	A nice RPN calculator prog, supports calculations with binary, octal, decimal, hex, float, and complex numbers. Other features include 32 registers for storing data and transcendental functions. V1.0, includes source. Author: Steve Bonner



& Public Domain Software

Mach	"mouse accelerator" prog. also includes hotkeys, features of sun mouse, clicktofront, and popcli, a title bar clock with a bbs online charge accumulator, and more. V1.6a, includes source. Author: Brian Moats
PatEdit	A pattern editor for creating patterns to input to the Amiga SetAfPt macro call. This call sets the area fill pattern for the area filling graphics (RectFill, AreaDraw, etc). Includes source. By Don Hyde
OMan	Mandelbrot generator written partially in assem. for speed. Includes source. By Steve Bonner

Fred Fish Disk 131

Dfc	Copies disks like Maurader, but multitasks. Replaces both diskcopy and format but is smaller than either. It even has a nice little Intuition interface. Includes source. Author: Tomas Rokicki
HyperBase	Shareware database management system. V1.6, binary only, source available from authors. Update of FF58. By: Michael MacKenzie, Marc Mengel, & Craig Norborg
Life	A new version of Tomas's ancient Life game, but with a new macro language for setting up patterns, some good examples, and some more good stuff. Includes source Author: Tomas Rokicki
Mackie	A Popcli replacement that draws pretty lines on the screen in blanking mode. Includes source. Author: Software Distillery; enhancements by Tomas Rokicki
Mg1b	A version of Mg1b with an ARexx port and other improvements by Tomas Rokicki. Now define macros and bind them to function keys in your startup file. Includes source. Author: Various; enhancements by Tomas Rokicki
WFrags	Another version of Frags, this pops up a little window that updates occasionally. Necessary for developers who wonder what their program is doing to memory. Includes source. Author: Tomas Rokicki

Fred Fish Disk 132

Berserk	Animation, a "must see" for every Amiga user, and ranks with "Juggler" as a premier demo for the Amiga. The difference between this distribution, and FF100, is that this one includes "source". I.E. It contains all the object descriptions necessary to recreate the animation, Modify it, or use it as an example for creating your own animations. Fred Fish felt it was appropriate to have at least one animation that was available at the "source code" level. Author: Leo Schwab

Fred Fish Disk 133

Conman	Extremely useful Shareware replacement for the standard console handler, provides line editing and command line histories completely transparent to any application program that uses CON: windows. Version 1.1, binary only, updateof FF100. New features include additional editing keys, fast search keys, undo key, clear history command, and more. Author: William Hawes
Crc	Two programs useful for generating 16-bit CRC listings of the contents of disks, and verifying that a given disk's files still compute to the same CRC's as listed. V1.0, binary only. Author: Don Kindred
CrcLists	Complete CRC check files for disks 1-128 of the library, using the Crc program also included on this disk. These were made directly from Fred's master disks. Author: Fred Fish
Overscan	Patches the Intuition library so that sizable windows with MaxHeight of 200 (400 in interlace) and screens with Height of 200 (400 in interlace) will take advantage of the PAL overscan capability of Intuition V1.2. This seems to be useful only for European users that wish to run software written for the American market, without modifying the applications, but still using the additional space. Includes source. Author: Ari Freund

Fred Fish Disk 134

BoingThrows	50 frame HAM animation done with Sculpt-3D, and DigiPaint. The animation took about 325 hours of runtime to generate. By Marvin Landis
Browser	A workbench tool, using text-only windows, that makes all files in the system accessible for executing, copying, moving, renaming, deleting, etc. Billed as a "programmers workbench". Version 1.2, binary only. Author: Peter da Silva
Dme	V1.29 of Matt's text editor. Simple WYSIWYG editor designed for programmers. Not a WYSIWYG word processor in the traditional sense. Features include arbitrary key mapping, fast scrolling, title-line statistics multiple windows, and ability to iconify windows. Update of FF113, includes source. By Matt Dillon
Find	Utility searches for files that satisfy a given boolean expression of attributes, starting from a root pathname and searching recursively down through the hierarchy of the file system. Very much like the Unix find program. V1.0, includes source. By Rodney Lewis
Library	Demo version of a shareware program that stores textual information without regard to structure or content, and allows complicated searching for specific patterns. Written in assembler for speed, binary only. Author: Bill Brownson
SmartIcon	Shareware Intuition objects iconifier. V1.0 is limited to iconifying windows, adds a new "iconify gadget" to each window, that when clicked, iconifies the window into an icon in the ram: disk. Binary only, source available from author. Author: Gauthier Groult

Fred Fish Disk 135

TeXF	A selection of 78 TeX fonts, with a conversion program to convert them to Amiga fonts. 22 different fonts at various sizes, ranging from 15 pixels high to more than 150 pixels. The conversion program can also be used with the fonts distributed with AmigaTeX, yielding an additional 1000 or so more fonts for use with other Amiga programs. V2.5, binary only. By: Ali Ozer

Fred Fish Disk 136

AsmToolBox	Assembler "toolbox" created to make interfacing between assembler programs and AmigaDOS easy. With source. By Warren Ring
Bison	A replacement for unix "yacc" command. From the GNU (GNU is Not Unix) effort. It is a port of the latest GNU version, done by William Loftus, with the goal of preserving all of bison's current features. Includes source and testing program "calc". Author: Bob Corbett and Richard Stallman.
Iff2Pcs	Interactive puzzle program takes any IFF file containing up to 16 colors, and breaks it up into squares to make a puzzle which the user can then piece back together again. V1.1, update of FF122, includes source. Author: Ali Ozer
Paste	Version of the Unix paste utility. Paste concatenates corresponding lines of the specified files into a single output line (horizontal or parallel merging) or concatenates them into alternate lines (vertical or serial merging). Includes source. Author: David Ihnat
YaBoingII	A game program demonstrating hardware sprite usage, including collision detection. Update of FF36. Includes source. Author: Ali Ozer, based on original by Leo Schwab
Zoo	File archiver, much like "arc" in concept, but different in implementation and user interface details. Includes some nice features that "arc" lacks (such as file/path names up to 255 characters in length). This is version 1.71, update of FF108. Binary only. Author: Rahul Dhesi, Amiga port by Brian Waters

Fred Fish Disk 137

Ct	Program to display images from a CT scanner, along with several interesting sample images of scans of real people, including a skull, brain, heart, and spine. Each image is 256 by 256 pixels in 2048 gray scale. The display software, though it has a primitive user interface, is quite powerful, including functions like convolutions, averaging, laplacians, unsharp masking, edge detection, gradients, etc. Binary only. Author: Jonathan Harman
JeansIcons	Miscellaneous cute icons created for AMUC's monthly newsletter disk. Submitted by Stephen Vermeulen. Author: Steve Jeans
Muncho	A cute little program which plays a digitized sound sample when you insert or remove a disk from your drive. If you don't like the sounds, you can replace them with your own. Binary only. By: Andrew Werth
Sit	Update to the Set Icon Type prog. on FF107. V1.10, includes source. Author: Stephen Vermeulen
VGad	A new gadget editor that takes two pictures of the window and its gadgets, one being the normal gadget state and the other being the fully selected state, then merges the data and converts to C source code. V1.0, binary only. Author: Stephen Vermeulen
VirusX	A boot sector virus check program that runs in the background and automatically checks all inserted disks for a nonstandard boot sector. Such disks can optionally have their boot sector rewritten to remove the virus. Includes source. Author: Steve Tibbett
VLabel	Program to print fancy customized disk labels. It will combine an IFF picture and up to 50 lines of text (which may be placed arbitrarily in any font or point size) then print the result. The IFF picture can be virtually any size (up to 1008 by 1000). It will also print labels from a batch file produced by SuperBase. V1.20, binary only. Author: Stephen Vermeulen

Fred Fish Disk 138

AmigaLine	A series of various technical notes for Amiga programmers. Author: Byrce Nesbitt
Diff	Program that uses the same algorithm as the Unix diff program and also produces context diffs, suitable for use with patch. Binary only. Author: Unknown (Decus C diff?)
Foreach	A simple but useful program that expands a wild card file specification and then invokes the specified command once per expanded filename, with the expanded filename as the command argument. Includes source. Author: Jonas Flygare
MacFont	A conversion tool to convert Mac fonts to Amiga fonts. Binary only. Author: John O'Neill and Rico Mariani
ModulaTools	Various useful routines for those programming in Modula on the Amiga. Update to version on disk 94, includes source. Author: Jerry Mack
Vt100	Two new versions of Dave's vt100 terminal emulator. One version, based on vt100 2.5, has been enhanced by John Barshinger to include an iconify feature, add full 132 column support using overscan, and other miscellaneous features (binary only). The second version is release 2.8 of the mainstream version of vt100, as enhanced and supported by Tony Sumrall. This one includes source. Author: Dave Wecker

To Be Continued.....

In Conclusion

To the best of our knowledge, the materials in this library are freely distributable. This means they were either publicly posted and placed in the public domain by their authors, or they have restrictions published in their files to which we have adhered. If you become aware of any violation of the authors' wishes, please contact us by mail.

IMPORTANT NOTICE!

This list is compiled and published as a service to the Commodore Amiga community for informational purposes only. Its use is restricted to non-commercial groups only! Any duplication for commercial purposes is strictly forbidden. As a part of Amazing Computing™, this list is inherently copyrighted. Any infringement on this proprietary copyright without expressed written permission of the publishers will incur the full force of legal actions.

Any non-commercial Amiga user group wishing to duplicate this list should contact:

PiM Publications, Inc.
P.O.Box 869
Fall River, MA 02722

PiM Publications Inc. is extremely interested in helping any Amiga user groups in non-commercial support for the Amiga.

•AC•

Amaze Me

Please use this order form when subscribing to Amazing Computing™, ordering Back Issues, or ordering Amiga™ Public Domain Software.

Name____
Street____
City ______________________ **ST.**____ **Zip** __________
Amount Enclosed ________________

Please circle the appropriate item: New Subscription Renewal

Please start my subscription to Amazing Computing™ with the next available issue or renew my current subscription.
I have enclosed $24.00 for 12 issues in the U.S. ($36.00 Canada and Mexico, $44.00 foreign surface).
All funds must be in U.S. Currency drawn on a U.S. Bank.

Back Issues:
$5.00 each US, $6.00 each Canada and Mexico, $7.00 each Foreign Surface.

Please circle your Back Issue choices below:

Vol1.1	Vol1.2	Vol1.3	Vol1.4	Vol1.5	Vol1.6	Vol1.7	Vol1.8	Vol1.9	Vol2.1	Vol2.2	Vol2.3	Vol2.4
Vol2.5	Vol2.6	Vol2.7	Vol2.8	Vol2.9	Vol2.10	Vol2.11	Vol2.12	Vol3.1	Vol3.2	Vol3.3	Vol3.4	

Public Domain Software:
$6.00 each for subscribers (yes, even the new ones!)
$7.00 each for non-subscribers
(foreign orders must be a minimum of three disks)

Please circle your Public Domain Software choices below:

Amicus:

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A11	A12
A13	A14	A15	A16	A17	A18	A19	A20	A21	A22	A23	A24	A25
A26												

Fred Fish:

FF1	FF2	FF3	FF4	FF5	FF6	FF7	FF8	FF9	FF10	FF11	FF12	FF13
FF14	FF15	FF16	FF17	FF18	FF19	FF20	FF21	FF22	FF23	FF24	FF25	FF26
FF27	FF28	FF29	FF30	FF31	FF32	FF33	FF34	FF35	FF36	FF37	FF38	FF39
FF40	FF41	FF42	FF43	FF44	FF45	FF46	FF47	FF48	FF49	FF50	FF51	FF52
FF53	FF54	FF55	FF56	FFNA	FF58	FF59	FF60	FF61	FF62	FF63	FF64	FF65
FF66	FF67	FF68	FF69	FF70	FF71	FF72	FF73	FF74	FF75	FF76	FF77	FF78
FF79	FFNA	FF81	FF82	FF83	FF84	FF85	FF86	FF87	FF88	FFNA	FF90	FF91
FF92	FF93	FF94	FF95	FF96	FF97	FF98	FF99	FF100	FF101	FF102	FF103	FF104
FF105	FF106	FF107	FF108	FF109	FF110	FF111	FF112	FF113	FF114	FF115	FF116	FF117
FF118	FF119	FF120	FF121	FF122	FF123	FF124	FF125	FF126	FF127	FF128	FF129	FF130
FF131	FF132	FF133	FF134	FF135	FF136	FF137	FF138					

(NA denotes disks removed from the collection)

Please complete this form and mail with check or money order to:

Amazing Computing
PiM Publications, Inc.
P.O.Box 869
Fall River, MA 02722

Please allow 4 to 6 weeks for delivery
All checks should be in US Funds drawn on a US bank

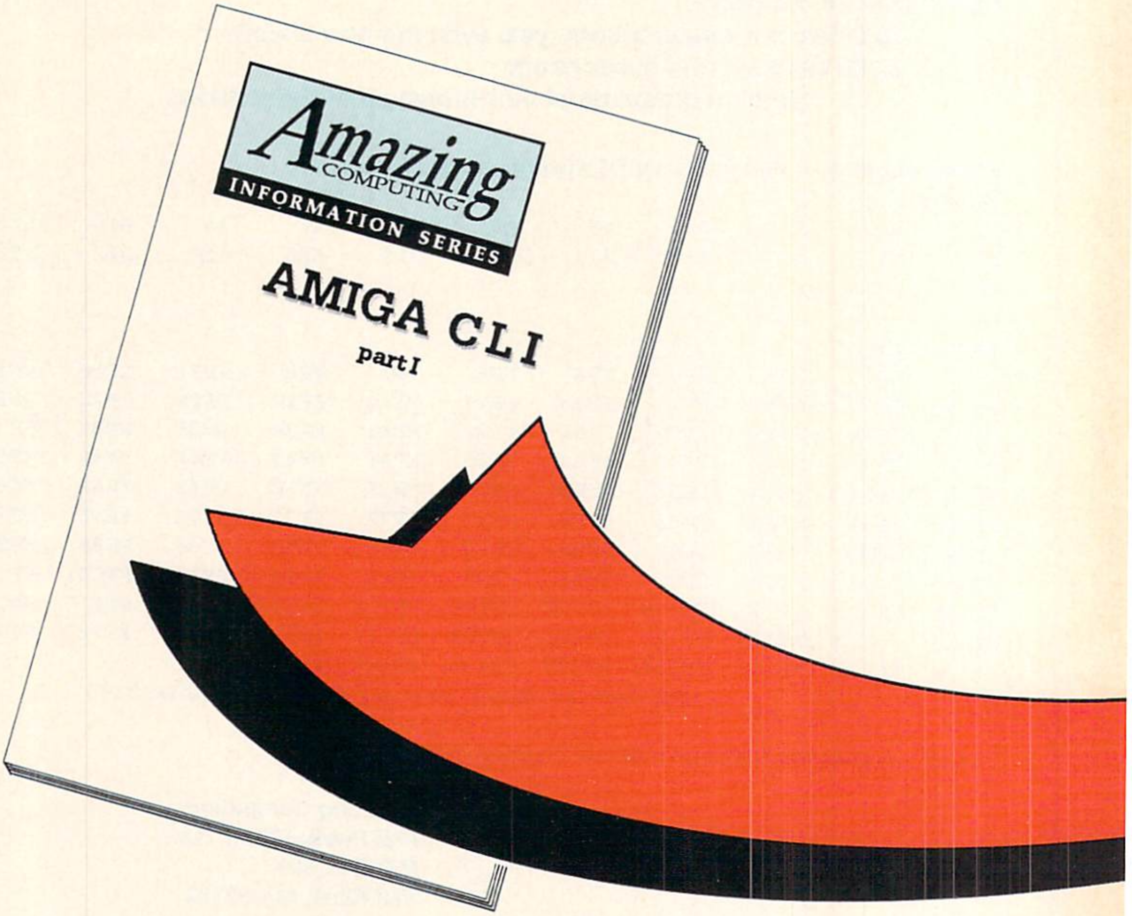
Two New Amiga

Introducing the new Amazing Computing Reader Service

For the first time ever, Amazing Computing is offering even more information to its readers through an Index of Advertisers/Reader Service Card. If any of the products advertised in this issue of Amazing tweak your interest, simply circle the company name on the form below and mail it back to us. (If you don't want to lose even a bit of the info already in the magazine, you can photocopy the form)

Because This Is Amazing Computing!
Not only will you receive in depth information about the products you indicate, but you will also receive a free copy of *Amiga CLI, part 1*, the first installment in the new Amazing Computing Information Series. This continuing series of handy reference booklets will provide quick, crucial information about the Amiga topics you are faced with most often. Once you have these booklets at your side, you'll wonder how you ever got by without them.*

Please take a few seconds to fill out the Reader Service Card ... and get even more information from the magazine that has always given you the most!



Sign up now for Amiga info and a **FREE GIFT!** *

---

* *AMIGA CLI, part 1* is available in limited quantities, while supplies last.

Sources Of Information!

Card and The Amazing Computing Information Series!

Amazing Computing's First Index of Advertisers/ Reader Service Card

Please print your name and address below, circle any advertisers that interest you, and send to:

Reader Service
PiM Publications, Inc.
P.O. Box 869
Fall River, MA 02722

Name ____

Address ______________________________________

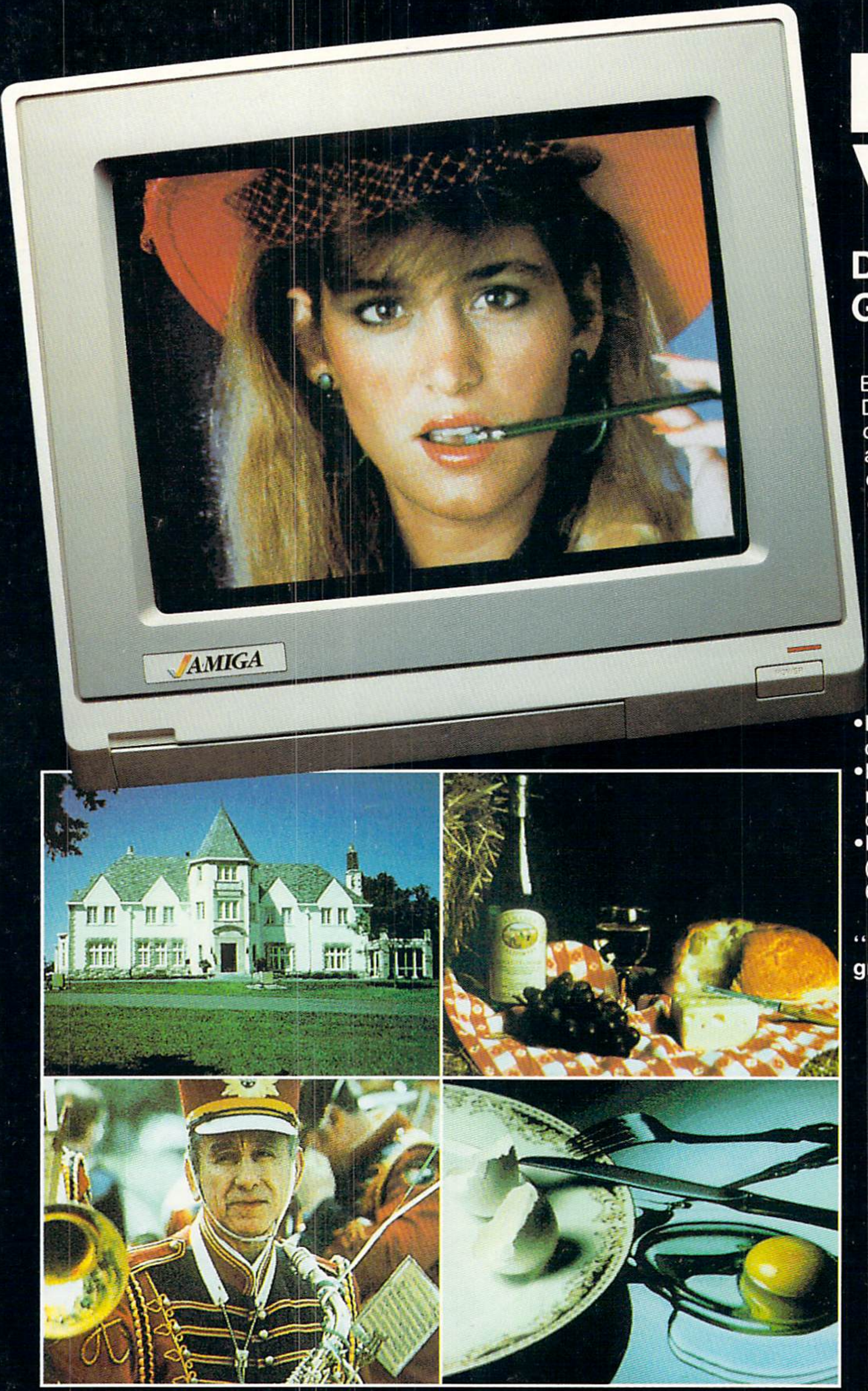
City _____________________ State ________ Zip ________

☐ Please send my free copy of Amazing Computing's Information Series *AMIGA CLI, part 1* * (one per household, one per request).

A-Squared Distributions, Inc.	9
Actionware	1
Amazing Computer Systems, Inc.	101
Ami Expo	119
Central Coast Software	11
Century Systems	104
Comp-U-Save	95
Computer Mart	93
CV Designs	118
D-Five Associates	106
Datasound	30
Delphi Noetic Systems, Inc.	19
Digital Wizards, Inc.	99
Digitronics	97
Don's Computer Works	98
Eraware	22
Fuller Computer Systems, Inc.	44
Golden Gate Shows	57
HyperTek/Silicon Springs	108
Jumpdisk	96
KETEK	110
Lamplighter Software, Inc.	21
Lattice, Inc.	7
Logical Design Works	41
Lynn's Luna C	45
Magnetic Images Co.	84
Megatronics	CII
MicroBotics, Inc.	56
Microsmiths, Inc.	91
NewTek	CIV
Newwave Software	31, 100
Peacock System	42
Pioneer Computing	105
Rittinghouse Software Development Co.	90
Sedona Software	38
Software Advantage Consulting Corp.	46
Spencer Organization	35, 107
Spirit Technology	57
SunRize Industries	15
Syndesis	87
Telecom Soft	23, 39
The Memory Location	86
The Other Guys	3
The Right Answers Group	94
TRU-IMAGE	116
West Coast Telecom	85

Please allow 6 to 8 weeks for delivery. A photocopy of this form is acceptable.

**AMIGA CLI, part 1* is available in limited quantities, while supplies last.



DIGI VIEW

DELIVERS ULTIMATE GRAPHICS POWER

Bring the world into your Amiga with Digi-View, the 4096 color video digitizer. In seconds you can capture any photograph or object your video camera can see in full color and with clarity never before available on a home computer. Digi-View's advanced features include:

- Dithering routines give up to 100,000 apparent colors on screen
- NewTek's exclusive Enhanced Hold-and-Modify mode allows for exceptionally detailed images
- Digitize images in any number of colors from 2 to 4096
- Print, animate, transmit, store, or manipulate images with available IFF compatible programs
- Digitize in all Amiga resolution modes (320x200, 320x400, 640x200, 640x400)

"Digi-View sets new standards for graphics hardware"-*InfoWorld*

Digi-View is available now at your local Amiga dealer or call:
1-800-843-8934

ONLY $199.95

NEWTEK
INCORPORATED